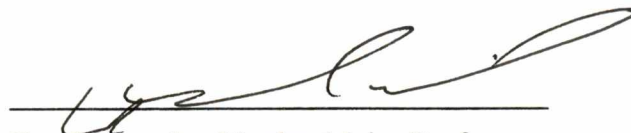


To the Graduate Council:

I am submitting herewith a thesis written by David Ray Michael Jr. entitled "Design of a Cost-Optimal Solenoid." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science, with a major in Engineering Science.

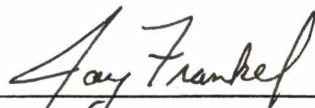


Dr. Christopher Pionke, Major Professor

We have read this thesis and
recommend its acceptance:



Dr. Frank Speckhart



Dr. Jay Frankel

Accepted for the Council:



Associate Vice Chancellor and
Dean of the Graduate School

DESIGN OF A COST-OPTIMAL SOLENOID

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

David Ray Michael Jr.
December 1998

DEDICATION

This thesis is dedicated to my parents,
Mr. David R. Michael Sr. and Mrs. Judy K. Michael,
who have given me the love and opportunities that have allowed me to succeed.

ACKNOWLEDGEMENTS

This work has proven to be a unique and challenging experience and I am particularly grateful for the help provided by a number of individuals. First, I would like to thank Dr. Chris Pionke who has acted as my Major Professor, advisor, and committee member. Aside from providing invaluable academic advice and assistance, Dr. Pionke has been a great friend and true mentor. I would also like to thank my committee members, Dr. Jay Frankel and Dr. Frank Speckhart, for their assistance and suggestions that have made this project a success. Finally, I would like to thank my best friend, Nikki, whose love and encouragement have help me reach my goals.

ABSTRACT

This study outlines the development of cost-optimal solenoid designs. A cost-optimal solenoid is defined here as that which meets given performance requirements at the lowest possible cost. Performance requirements for the solenoid are specified as a minimum force to be delivered by the solenoid armature at a specific position along the armature stroke distance.

A major component of this optimization study involves the use of the finite element method as a means for characterizing the effect of design variables on solenoid performance. Since the solenoid is nonlinear in its behavior, the finite element method is used to numerically solve for the force delivered by the armature of the solenoid. The method of least squares is then used to curve fit a force function to the finite element results. A second function is developed that expresses solenoid cost as a function of the same variable set. Minimization techniques are then used to minimize cost and maximize performance, thus identifying the cost-optimal design configuration.

It has been concluded that this study is effective in producing cost-optimal solenoid designs. In order to assess the amount of confidence that is given to the resulting designs, the individual steps of the study are examined for accuracy. By verifying the accuracy of each step of the optimization process, it is concluded that the methodology, as a whole, is accurate and effective.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
Problem Statement	2
Basic Mathematical Principle	3
Background of Related Studies	5
Solenoid Force Analysis and Optimization	7
II. SOLENOID DESCRIPTION AND ANALYSIS	9
Introduction	9
Solenoid Description	10
Physics of the Solenoid	12
Analytical Methods for Determining Solenoid Performance	16
Example Finite Element Problem	21
III. OUTLINE OF METHODOLOGY	27
Introduction	27
Design Variables	28
Two Phases of Study	29
Phase One.	31
Phase Two.	31
Assumptions	32
Methodology Summary	33
IV. FINITE ELEMENT ANALYSIS OF THE SOLENOID	35
Introduction	35
Convergence Study	42
Creating a Database	44
Phase One Analysis	45
Phase Two Analysis	49
V. METHOD OF LEAST SQUARES	52
Introduction	52
Simplified Example	53
The Coefficient of Determination	59
VI. PHASE ONE OPTIMIZATION	62
Introduction	62

Developing a Force Function	62
Assessing the Quality of the Curve Fitting Function	68
Phase One Optimization	69
Further Analysis	73
Applying Information	74
 VII. PRESENTATION OF COST EQUATIONS	 76
Washer Costs.	77
Steel Casing Cost	77
Core and Plunger Cost	78
Coil Cost.	79
Total Solenoid Cost	79
 VIII. PHASE TWO OPTIMIZATION.	 81
Introduction.	81
Developing a Force Function	82
Introduction to Phase Two Optimization	90
Penalty Method.	91
The Actual Optimization Problem	93
Search Algorithm	94
 IX. DESIGN EXAMPLES	 97
Example One	97
Example Two.	99
Example Three	100
 X. CONCLUSIONS	 102
 LIST OF REFERENCES	 107
 APPENDICES	 110
Appendix A: Example Problem COSMOS Finite Element Input File	111
Appendix B: Small Push-Pull Solenoid COSMOS Finite Element Input File.	115
Appendix C: Low Reluctance COSMOS Finite Element Input File	119
Appendix D: Phase One Finite Element Results.	123
Appendix E: Phase Two Finite Element Results.	127
Appendix F: Maple Worksheet / Example Least Squares Problem.	132
Appendix G: Maple Worksheet / Phase One Calculations	135
Appendix H: Maple Worksheet / Cost Equations	142
Appendix I: Maple Worksheet / Altered Curve Fit Function Calculation	145
Appendix J: Maple Worksheet / Phase Two Calculations	150

Appendix K: Fortran Code Phase Two Search Algorithm.	157
Vita.	163

LIST OF TABLES

TABLE	PAGE
2.1 Relative Permeabilities of Common Materials	14
2.2 B-H Curve Data Points for COSMOS Verification Problem	24
4.1 Finite Element and Experimental Results	39
4.2 Modified B-H Data Points	40
4.3 Finite Element and Experimental Results	41
From Modified B-H Curve	
4.4 Geometric Range of Notch Depth and Notch Angle	47
4.5 Example of Phase One Analysis Results	49
4.6 Geometric Ranges of Phase Two Design Variables	51
4.7 Example of Phase Two Analysis Results	51
5.1 r^2 Coefficient Values for Various Degree Curve Fitting Function	61
6.1 Results of Phase One Random Tests	69
8.1 Results of Phase Two Random Tests	89
9.1 Summary of Design Example One Results	98
9.2 Summary of Design Example Two Results	99

LIST OF FIGURES

FIGURE	PAGE
1.1 Example of Performance vs. Design Parameter	3
1.2 Example of Cost vs. Design Parameter	4
2.1 Small Push-Pull Solenoid	10
2.2 Cross-Section of the Small Push-Pull Solenoid	12
2.3 Magnetic Field of a Coil Depicted by Lines of Force	13
2.4 B-H Curve Depicting Nonlinear Permeability of a Material	15
2.5 Magnetic Structure of a Simplified Solenoid	17
2.6 Fringing Effects in Air Gap	19
2.7 Simplified Equivalent Expression for the Magnetic Circuit of a Solenoid	20
2.8 Example Problem Solenoid	21
2.9 Keypoints and Curves of Example Finite Element Problem	23
2.10 Finite Element Mesh of Example Problem	24
2.11 Example of COSMOS Output File Showing Force on Plunger	26
3.1 Design Variables of the Solenoid	29
3.2 Assumed Dimensions and Design Variables	33
3.3 Flow Diagram Showing Steps Used to Produce a Cost-Optimal Solenoid	34
4.1 Finite Element Model of the Small Push-Pull Solenoid	36
4.2 Experimental Test Data Showing Force vs. Stroke Distance	38

4.3	Experimental and Finite Element Results with Modified B-H Curve	38
4.4	Example Problem and Modified B-H Curves	40
4.5	Experimental and Finite Element Force Plots From Modified B-H Curve	41
4.6	Convergence Study Results	43
4.7	Phase One Design Variables	45
4.8	Conical Pole Interface Configuration(a) and Flat Faced Pole Interface. Configuration(b)	46
4.9	Low Reluctance Finite Element Model Used for Phase One Analysis	48
4.10	Phase Two Design Variables	50
5.1	Example Data Set	53
5.2	Plot of Data Set with Quadratic Curve Fitting Function	55
5.3	Plot of Data Set with 3 RD Degree Curve Fitting Function	57
5.4	Plot of Data Set with 4 TH Degree Curve Fitting Function	57
5.5	Plot of Data Set with 12 TH Degree Curve Fitting Function	58
6.1	Phase One Design Variables	63
6.2	Plot of Finite Element Results Showing Force vs. Notch Angle	65
6.3	Plot of Finite Element Results Showing Force vs. Notch Depth	65
6.4	Plot of Finite Element Results Showing Force vs. Stroke Position	66
6.5	Surface Plot of Force vs. Notch Angle and Notch Depth	70
6.6	Force vs. Notch Angle for Stroke Position of 0.04 inches	75
6.7	Force vs. Notch Angle for Stroke Position of 0.14 inches	75
8.1	Phase Two Design Variables	81
8.2	Plot of Finite Element Results Showing Force vs. Case Radius.	83

8.3	Plot of Finite Element Results Showing Force vs. Length.	84
8.4	Plot of Finite Element Results Showing Force vs. Plunger Radius	84
8.5	Plot of Finite Element Results Showing Force vs. Case Wall Thickness.	85
8.6	Random Test #1 Results Showing Force vs. Stroke Position.	87
8.7	Random Test #2 Results Showing Force vs. Stroke Position.	88
8.8	Random Test #3 Results Showing Force vs. Stroke Position.	88
8.9	Two Dimensional Surface Plot of a Penalty Function	94
8.10	Flow Diagram For Fortran Search Algorithm	96
9.1	Results From Design Example One	97
9.2	Results From Design Example Two	99
9.3	Results From Design Example Three	101

LIST OF SYMBOLS

Φ	Magnetic Flux
B	Magnetic Flux Density
H	Magnetic Field Intensity
I	Electric Current
N	Number of Turns
$\mathfrak{S}$	Magnetomotive Force
μ	Magnetic Permeability
μ_o	Relative Magnetic Permeability
$\mathfrak{R}$	Reluctance
$\mathfrak{R}_g$	Reluctance Due To Air Gap
$\mathfrak{R}_t$	Total Reluctance
A_c	Cross-Sectional Area
A_g	Cross-Sectional Area of Air Gap
V	Electric Voltage
R	Electric Resistance
F	Solenoid Force Output
S	Stroke Position
A	Notch Angle
D	Notch Depth
C_R	Solenoid Case Radius

L	Solenoid Length
P_R	Solenoid Plunger Radius
W	Case Wall Thickness
η	Least Squares Error
$\hat{y}$	Interpolated Value of y
$\bar{y}$	Average Value of y
r^2	Coefficient of Determination

CHAPTER I

Introduction

A particular dictionary defines engineering as “the art or science of making practical application of the knowledge of pure sciences such as physics, chemistry, biology, etc” [17.]. By making “practical” use of scientific knowledge, it is implied that an engineer’s job is to create the best or optimal solutions to given problems. Design optimization is a fundamental goal of virtually every engineer who strives to create a device or system that must meet a specified requirement. Unfortunately, fulfillment of this goal has suffered because there are few organized methods available to help engineers optimize their designs [6]. The infinite range of problems faced by engineers makes it difficult to develop a clear-cut “recipe” for producing an optimal solution.

With regard to mechanical design, optimization approaches are routinely used to determine the parameters of a component so as to reduce the overall cost [15]. More specifically, the objective of the designer is to produce a cost-optimal design that satisfies a set of functional performance requirements. For given performance specifications, the wide array of parametric combinations to choose from often poses a bewildering task of selection. Consequently, design practice often relies on experienced-based judgement (heuristics) of seasoned designers, aided to an extent by theoretical models [14].

As an alternative to heuristics, methods are being explored that formulate mechanical design tasks as optimization problems [15]. With such methods,

combinatorial optimization occurs in parallel with the design process so that the final design is driven toward an optimal configuration. Research in the field of combinatorial optimization is aimed at developing efficient techniques for finding minimum or maximum values of a function of many independent variables [9]. With respect to mechanical design, mathematical techniques are being researched that minimize cost functions of several design variables while maintaining performance requirements in the form of constraint equations.

Problem Statement

This study illustrates the use of optimization techniques in the mechanical design of solenoids. These optimization techniques are used to set final geometric parameters of the solenoid design so as to minimize cost while meeting a specified output requirement. This study also describes the use of the finite element method as a necessary tool for handling the inherent nonlinearities that arise in describing solenoid performance. A database of finite element analysis results is used to create an explicit mathematical formulation of solenoid performance with respect to a number of design variables. A second component of this study involves the development of the mathematical expression of solenoid cost with respect to the same design variable set. Lastly, minimization techniques are used to define the cost-optimal solenoid design. The cost-optimal solenoid is defined as that which achieves a minimum performance requirement at the lowest possible cost.

Basic Mathematical Principle

To further explain the basic mathematical principle behind this study, consider a simple example. Suppose that the performance, P , of a particular product is plotted as a function of a single design parameter, X , as in Figure 1.1. Also, suppose that the minimum performance requirement for this product has the value M . Notice that the performance requirement is met or exceeded over a finite span of X . There are two values of X that exactly produce the minimum performance requirement; X_L and X_R .

Now suppose that the cost to produce the product is plotted as a function of the same design parameter, X . Assume that the cost linearly increases with X as shown in Figure 1.2. By examining both plots, it is obvious that X_L is the cost-optimal value for X since it satisfies the minimum performance requirement at the lowest cost. Although this is a simple example, it represents the basic idea behind the design optimization

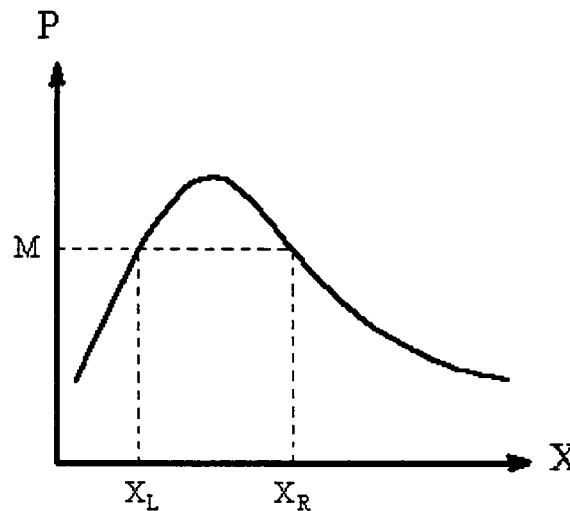


Figure 1.1. Example of Performance vs. Design Parameter

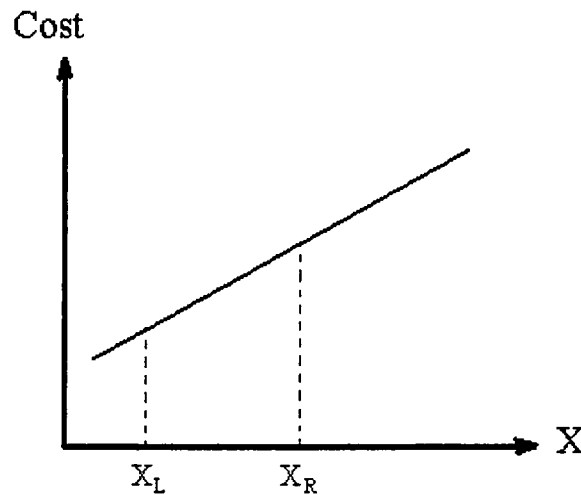


Figure 1.2. Example of Cost vs. Design Parameter

methodology. The difficulty is that in real design problems, performance and cost are functions of many variables and the mathematics of cost optimization become complicated and difficult to solve.

Background of Related Studies

Optimization strategies have already been applied to a host of engineering design problems. These applications include the optimization of drive trains for hand-held drills [15], the design of cost-optimal composite beams [2], the optimization of concrete structures [14] and optimization in electronic packaging [7]. In fact, combinatorial optimization has been the focus of computer scientists and engineers for many years. One of the first papers on the subject was published in 1973 and introduced the power of

digital computers for solving optimization problems[9]. Since then, there have been a large number of studies involving the mathematics, numerical solution, and practical applications of combinatorial optimization [2,8,15]. Despite a large background of optimization studies, no published works have been uncovered on the topic of solenoid optimization.

An obvious reason for the attention received by optimization research lies in potential for cost savings. In fact, it is difficult to find a study on combinatorial optimization that does not aim to minimize cost. Interestingly, the most classical combinatorial optimization problem has nothing to do with cost (in the literal sense). It is referred to as the traveling salesman problem and serves as a classic example is because it is so simply stated. Given a list of N cities and a means for calculating the distance between them, one must plan the salesman's route, which will pass through each city once and return to the starting point, while minimizing the total distance traveled.

A particular paper uses the traveling salesman problem to illustrate the complex mathematics behind combinatorial optimization [9]. It is noted in this paper that all exact methods known for determining an optimal route require a computing effort that increases exponentially with N , so that in practice, exact solutions can be attempted only on problems involving a few hundred cities or less. As a result, the main focus of optimization research has been on iterative techniques for solving problems of this class.

Most optimization studies are unique in that they employ innovative techniques to solve their respective problems. These technique usually come in the form of complex computer algorithms that are used to solve for the optimal design. Some examples of

these algorithms include the Constrained Variable Metric Algorithm and the Function to Form Recursive Annealing Design Algorithm (FFREADA) [3,15]. The common goal of these programs is to minimize a cost equation while maintaining performance requirements in the form of constraint equations. In both cases this is done by methodically iterating through various design configurations such that the minimum cost configuration is defined as quickly and with as much computational efficiency as possible. Both methods are relatively complicated and require a substantial amount of computation.

Before such optimization techniques can be employed, it is obvious that a performance function must be developed. In some cases, this could likely be the most challenging task in the optimization process. Consider the following example of a study on the design of cost-optimal cooling towers [3]. The cooling performance of the tower is written as a function of several geometric parameters. The construction and maintenance costs are then represented as a function of the same set of variables. The study illustrates the mathematical minimization process that is used to identify the cost-optimal cooling tower design.

Although the study was successful in obtaining the cost-optimal cooling tower design, there are difficulties with the method. To produce performance equations for the cooling tower, it is not surprising that fundamental relationships regarding heat transfer, fluid mechanics, and thermodynamics are used. Unfortunately, mathematical representations of all physical phenomena do not exist. In these cases, the designer must develop the mathematical model of product performance with respect to the necessary

design parameters. This can be especially difficult for nonlinear systems in which fundamental physical relationships do not exist. Such is the case with the problem of solenoid optimization.

This study is unique in that it focuses on the challenging task of developing a performance equation for the nonlinear solenoid. In doing so, the finite element method is introduced as an efficient means for characterizing solenoid performance. Once the performance function is developed from finite element results, basic optimization techniques are then used to identify the cost-optimal solenoid design.

Solenoid Force Analysis and Optimization

It is shown in the following chapter that the physics that governs solenoid performance is largely nonlinear. It is also shown in the next chapter that due to the nonlinearity of the solenoid, closed form analysis of solenoid performance requires many simplifying assumptions. These assumptions greatly reduce the feasibility of developing an accurate physical model. As a result, the finite element is used as a means for analyzing the solenoid.

Developed in the 1950s, the finite element method is most generally described as a method for solving boundary-value problems. Engineers developed the finite element method as a way to utilize the power of digital computers to perform structural analyses on aircraft [1]. Today, the method is has been extended to a number of engineering disciplines including fluid dynamics and heat transfer [1]. Since the finite element

method solves problems numerically, it is especially effective in analyzing nonlinear phenomena such as that of the solenoid.

In this study, the method essentially involves the description of solenoid geometry as a computer model which is analyzed for performance. More specifically, the force capability of each solenoid model is determined. By analyzing a number of models with varying geometry, a large matrix of data corresponding to solenoid geometry and force performance is obtained. The database of results is then used to characterize the effects of each design variables on performance. This is achieved by interpolating an algebraic function to the data set. Since all performance optimization is based on computational results, the finite element method can be thought of as the foundation of this design methodology.

As previously stated, the goal of this study is to provide a method for identifying a cost-optimal solenoid. It should be noted, however, that the methods outlined in this report are not limited to solenoid design. The solenoid simply serves as a practical vehicle for illustrating the methodology of the study. Since the finite element method has a vast range of analysis capabilities, the optimization methods presented here have an equally vast range of application in mechanical design.

CHAPTER II

Solenoid Description and Analysis

Introduction

While cost-optimization is an important part of almost any design problem, there are a number of specific reasons why the solenoid serves as an appropriate example for illustrating the methodology of this study. First, the solenoid exhibits a number of design variables that may be incorporated into the design optimization process. Second, as shown in this chapter, the solenoid is largely nonlinear, thus making the development of an analytical model difficult and impractical. Because of this nonlinearity, the usefulness of the finite element is demonstrated. Last, the use of the finite element method to perform the necessary electromagnetic analysis illustrates the broad spectrum of finite element capabilities.

For the purpose of this study, solenoid performance is based solely upon the force delivered by the armature. An obvious first step then is to develop a means for analyzing the workings of the solenoid. More specifically, a method must be identified for obtaining the force capabilities of the solenoid armature as a function of geometry and material properties. Secondly, a method must be developed to analyze the cost of a solenoid. Before the development of these methods, however, a discussion of solenoids is presented. The remainder of this chapter describes the basic physical characteristics of

the solenoid as well as the physics that govern its motion. The final section of this chapter describes the finite element method as a means for analyzing solenoids.

Solenoid Description

A solenoid is a type of actuator that uses an electric current to produce a magnetic field that, in turn, drives the mechanical motion of an armature. The type of solenoid analyzed in this study is called a small push-pull solenoid and is shown in Figure 2.1. The function of a solenoid is to provide a mechanical force over a specified range of motion of the armature. Although the solenoid may act to push or pull, it is always the electromagnetic *attraction* that drives its motion. A spring is commonly used to return the armature to its original open position.

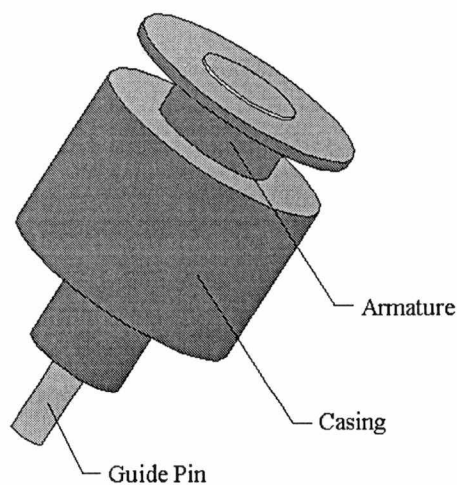


Figure 2.1. *Small Push-Pull Solenoid.*

The range of motion of the armature is referred to as the stroke distance of the solenoid. The performance of the solenoid is based on the force delivered by the armature over the stroke distance. The small push-pull solenoid is very cost-effective when small stroke distances are required and/or limited amount of electrical power is available [11]. A typical solenoid of this type may range from one to two inches in length and provide forces up to 40 lbs.

The armature of the solenoid has a washer at the end and moves along a brass guide pin. The small push-pull solenoid design utilizes a coil with the maximum amount of copper wire in the smallest amount of space. The coil assembly is packaged in a highly permeable steel casing to obtain the maximum amount of force with minimal size and weight.

A cross-sectional representation of a small push-pull solenoid is shown in Figure 2.2. As shown in the figure, the copper coil is encased in a thin nylon bobbin with a layer of tape on the outer edge. At the center of the solenoid is a steel core above which is the steel armature, also called a plunger. The conical pole interface configuration shown in Figure 2.2 (plunger-core interface) is common in the small push-pull solenoid design [11]. A thin polymer air gap spacer is seated between the casing and plunger. By preventing mechanical contact between the plunger and casing, unwanted residual magnetism is avoided [11].

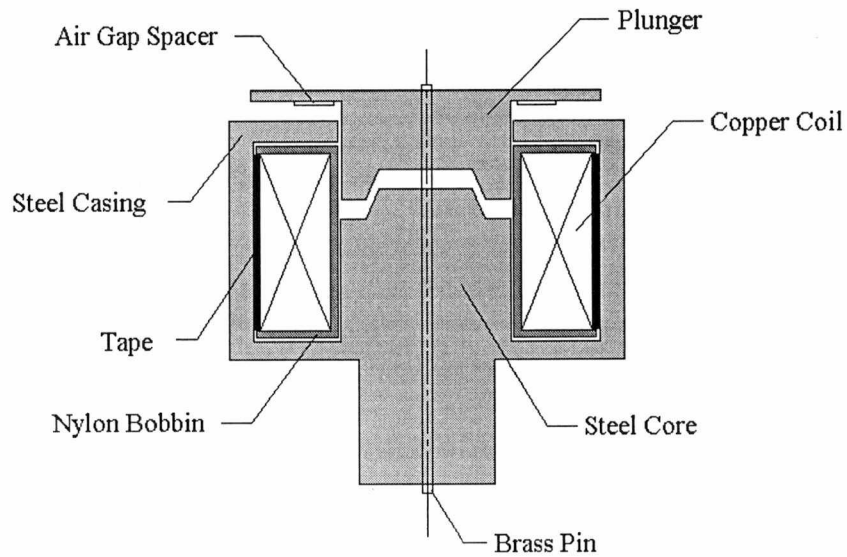


Figure 2.2. Cross-Section of the Small Push-Pull Solenoid.

Physics of the Solenoid

The magnetic field within the solenoid forms the connection between electrical and mechanical energy. Magnetic fields are generated by electric charge in motion. As charge moves through the copper coil of a solenoid, the coil becomes polarized. The resulting magnetic field may be depicted by means of lines of force that exit at the north pole of the coil (by definition) and enter at the magnetic south pole (Figure 2.3). The quantities used to describe the strength of a magnetic field are the magnetic flux, Φ , in units of webers (Wb); the magnetic flux density, $\mathbf{B}$ (Wb/m^2), or teslas (T); and the magnetic field intensity, $\mathbf{H}$ in units of amperes per meter (A/m). The latter two of these are vector quantities, and thus have associated magnitudes and directions.

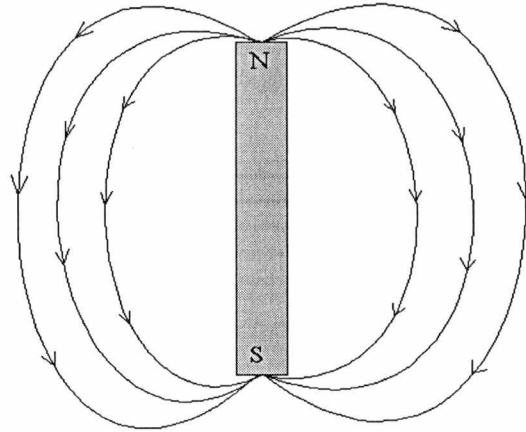


Figure 2.3. *Magnetic Field of a Coil Depicted by Lines of Force.*

The reason for introducing the magnetic field intensity is that it is independent of the properties of the materials that make up the magnetic circuit. Thus, a given magnetic field intensity, $\mathbf{H}$, will give rise to different flux densities in different materials. The source that governs the magnitude of the magnetic field is called the magnetomotive force. The field created by a single conducting wire is not very strong; however, if the wire is arranged into a tightly bound coil with many turns, the strength of the magnetic field is greatly increased. For a coil carrying a current, I , and having N turns, the associated magnetomotive force, $\mathfrak{F}$, is given by the equation,

$$\mathfrak{F} = N \cdot I \text{ (amp-turns)} \quad (2.1)$$

As previously stated, the magnetic flux densities within the components of the solenoid are material dependent. The relationship between the magnetic field intensity, $\mathbf{H}$, and the flux density, $\mathbf{B}$, is given by,

$$\mathbf{B} = \mu \mathbf{H} \text{ (Wb/m}^2\text{)} \quad (2.2)$$

where the parameter μ is approximated as a constant that is specific to a material and is called the permeability. Material permeability is often factored as the product of the permeability of free space, $\mu_0 = 4\pi \times 10^{-7}$ H/m, and the relative permeability, μ_r . Equation (2.2) can then be represented by,

$$\mathbf{B} = \mu_r \mu_0 \mathbf{H} \text{ (Wb/m}^2\text{)} \quad (2.3)$$

The size of μ_r represents a measure of the magnetic property of a material. The larger the value of μ_r , the smaller the current that is required to produce a large flux density in an electromagnetic structure. Consequently, the use of highly permeable steel in solenoids greatly enhances their magnetic properties. Table 2.1 gives values of μ_r for some common materials [13].

As previously stated, μ is often approximated as a constant for a particular material. In reality, the relationship between the magnetic flux density, $\mathbf{B}$ and the associated field intensity, $\mathbf{H}$, is characterized by the fact that the permeability of magnetic

Table 2.1. *Relative Permeabilities of Common Materials.*

M a t e r i a l	μ_r
A i r	1
P e r m a l l o y	100,000
C a s t S t e e l	1,000
S h e e t S t e e l	4,000
I r o n	5,195

materials is not a constant, but is a function of the magnetic field intensity. Also, all magnetic materials exhibit a phenomenon called saturation whereby the flux density increases in proportion to the field intensity until it cannot do so any longer. Figure 2.4 illustrates the general behavior of all magnetic materials. Since the B-H curve shown in the figure is nonlinear, the value of μ (which is the slope of the curve) depends on the intensity of the magnetic field. As will be shown, it is difficult to account for this inherent nonlinearity analytically.

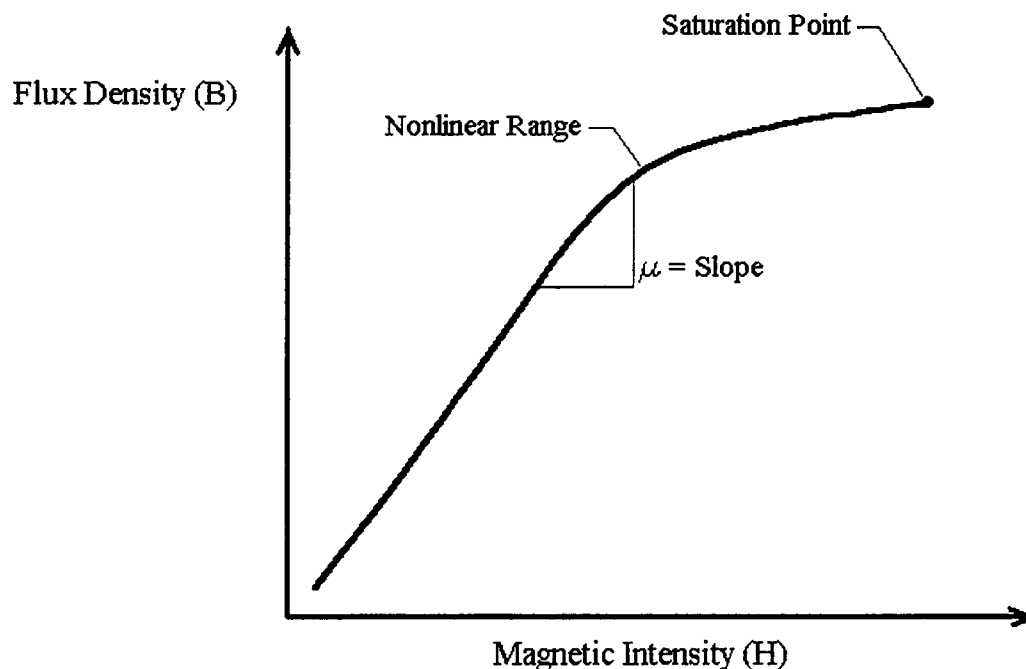


Figure 2.4. *B-H Curve Depicting Nonlinear Permeability of a Material.*

Analytical Methods for Determining Solenoid Performance

As an electric circuit diagram may be used to describe the movement of charge through a circuit, a magnetic circuit may be used to describe the magnetic flux through a solenoid. An equivalent magnetic circuit for a simplified solenoid is shown in Figure 2.5. A magnetic circuit can be analyzed with relations very similar to those used in electrical circuit analysis. For example, the magnetomotive force, $\mathfrak{F}$, may be viewed as analogous to the voltage source in an electrical circuit, and the flux, Φ , analogous to the electrical current. Also similar to an electrical circuit, the flux through a magnetic circuit has an associated resistance, $\mathfrak{R}$, which is given by,

$$\mathfrak{R} = \frac{L}{\mu A_c} \quad (2.4)$$

where L is the length of the component through which the flux travels, μ is the permeability, and A_c is the cross-sectional area of the component. This ‘resistance’ parameter is actually referred to as reluctance for magnetic circuits. The fundamental equation used to analyze magnetic circuits is given by,

$$\mathfrak{F} = \Phi \cdot \mathfrak{R} \quad (2.5)$$

which is analogous to Ohm’s law used in the analysis of electric circuits which is written as,

$$V = I \cdot R \quad (2.6)$$

where V is voltage, I is current, and R is the resistance.

Before a magnetic circuit can be used to analyze the magnetic flux within a solenoid and ultimately the force on the plunger, several simplifying approximations

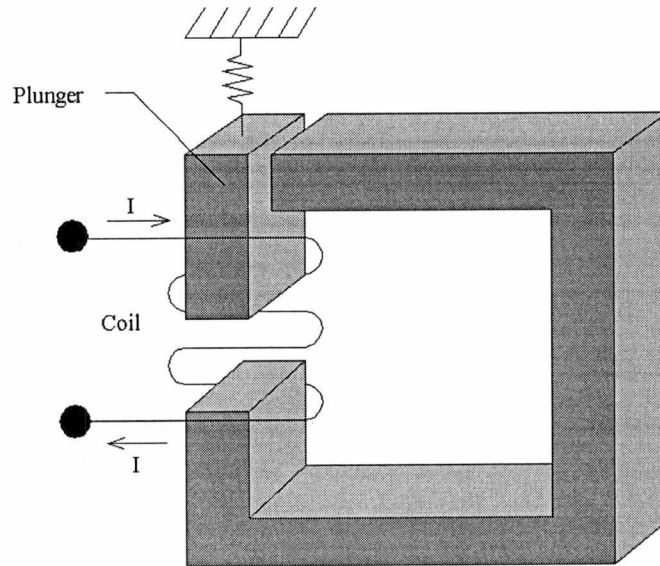


Figure 2.5. Magnetic Structure of a Simplified Solenoid.

must be made. The first of these approximations assumes that there exists a mean path for the magnetic flux, and that the corresponding mean flux density is constant over the cross-sectional area of the components of the solenoids. Because only a mean flux is calculated, material permeability must be approximated as a constant. This assumption thus approximates the nonlinear magnetic behavior of materials as linear. Lastly, the magnetic circuit model assumes that all flux remains confined to the magnetic components and as a result fails to compensate for flux leakage.

With these assumptions, the previously stated mathematical relationships may be used to analyze the flux through a simple solenoid (Figure 2.5). By computing the mean flux between the plunger and the core of the solenoid, the magnetic force that results can be obtained from the relation,

$$force = \frac{1}{2} \frac{\Phi^2}{\mu A_c} \quad (2.7)$$

where Φ is the mean flux through the solenoid, and A_c is the cross-sectional area of the air gap.

The solenoid structure has a second air gap at the plunger and steel case interface. The reluctance associated with each of these air gaps can be calculated by using the relation,

$$\mathfrak{R}_g = \frac{\delta}{\mu_o A_g} \quad (2.8)$$

where δ is the gap distance, μ_o is the permeability of free space, and A_g is the effective cross-sectional area of the gap.

Magnetic flux across an air gap displays a phenomenon known as fringing which is shown in Figure 2.6. Notice how the flux lines tend to spread out in the air gap while they remain straight and completely contained in the magnetic structure. The concept of constant flux density clearly breaks down in this case. To compensate for this, an effective gap area may be approximated with the relationship below,

$$A_g = (l + \delta)(w + \delta) \quad (2.9)$$

where l and w are the length and width that describe the cross-sectional area and δ is the air gap distance.

Before applying the relationships to solve for the force on the plunger of the solenoid, it is beneficial to write a simplified equivalent expression for the magnetic

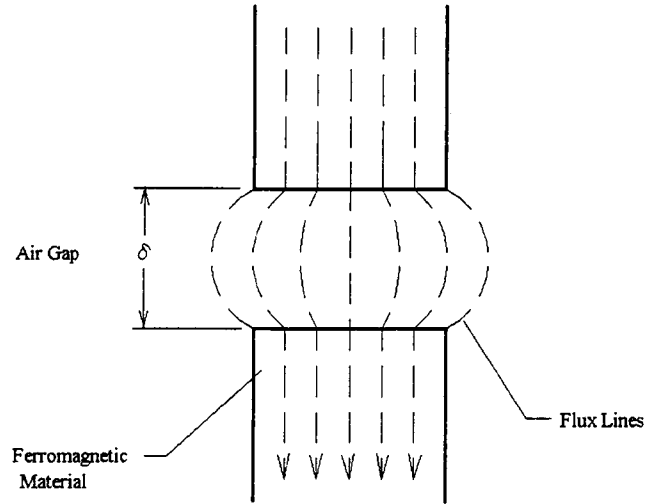


Figure 2.6. *Fringing Effects in Air Gap.*

circuit. Figure 2.7 shows an equivalent circuit with an associated reluctance for each component and air gap of the solenoid. The total reluctance, $\mathcal{R}_t$, of the system is equal to the sum of the reluctance of each of the components, which are determined from (2.5) and (2.8). By knowing the current, I , in the coil and the number of turns, N , the magnetomotive force can be determined from (2.1). The mean flux can then be determined by solving for Φ in (2.5) to get the relation:

$$\Phi = \frac{\mathfrak{F}}{\mathcal{R}_t} \quad (2.10)$$

After finding the mean flux, Φ , the force on the plunger may be calculated from (2.7).

Keeping in mind the major simplifying assumptions that were made, it is logical to assume that this model is only helpful in approximating the mean behavior of the magnetic circuit. It is also evident from this method that many important geometric details of the solenoid have approximated. For example, the method requires that the

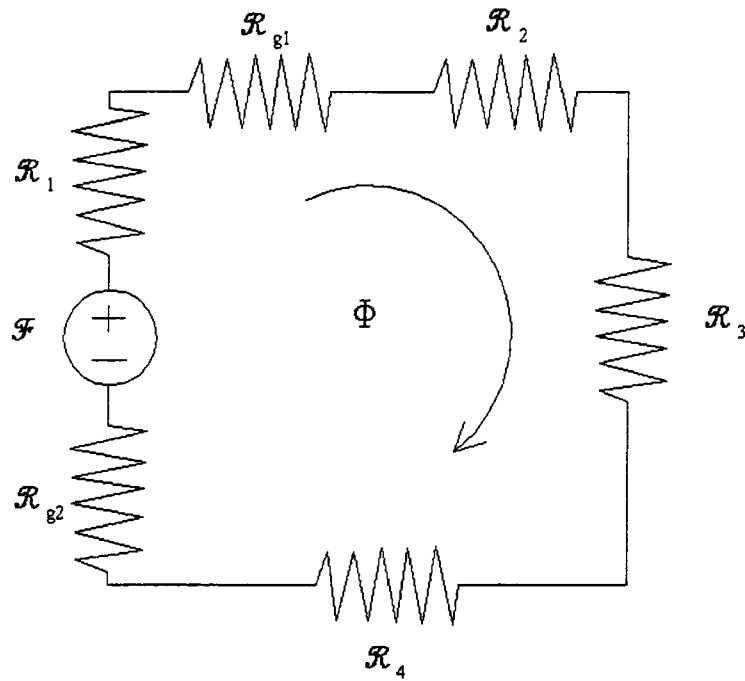


Figure 2.7. *Simplified Equivalent Expression for the Magnetic Circuit of a Solenoid.*

magnetic flux travel through constant cross-sections. As a result, calculating the force on the plunger in this way may offer only a crude estimate. This crudeness illustrates the need for a more accurate method of analyzing the performance of a solenoid.

While analytical methods are obviously impractical for accurately analyzing the solenoid, a numerical method may be valuable. To prove useful as an analysis tool, the method must account for the detailed geometry of the solenoid as well as the nonlinear magnetic behavior of its materials. The method must also produce results that closely correspond to empirical data that exists. It is shortly shown that this can be achieved with the finite element method described in the next section.

Example Finite Element Problem

As mentioned in Chapter I, the finite element method utilizes computers to numerically analyze physical phenomena. The particular finite element software used in this study is called COSMOS and features capabilities for solving electromagnetic problems. As an introduction to finite element analysis of the solenoid using COSMOS, this section describes the solution of an example problem that is provided in the COSMOS User's Guide [5]. This example problem outlines the modeling and analysis of a simple push-pull solenoid (Figure 2.8). The goal of the problem is to calculate the maximum flux and determine its location within the solenoid.

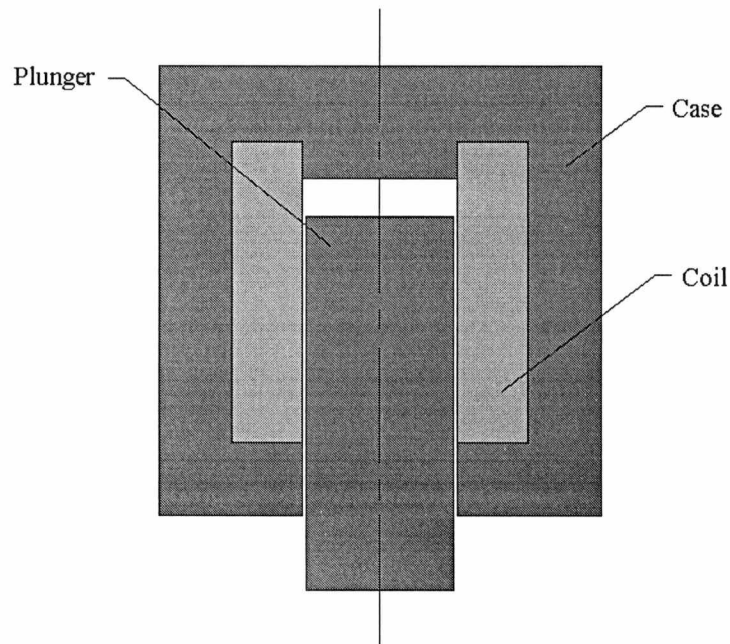


Figure 2.8. *Example Problem Solenoid.*

The finite element formulation for the example solenoid is simplified as a result of its axisymmetric geometry. The term axisymmetric means that the geometry of the solenoid is independent of the angle about the longitudinal axis. Because of this axial symmetry, the finite element model of the entire solenoid can be represented with a two-dimensional model of the solenoid's cross-section. Like the solenoid of the example problem, the small push-pull solenoid of this study is axisymmetric and utilizes a similar two-dimensional model.

The basic steps of performing the finite element analysis include creating the geometry of the solenoid, meshing the different material regions of the solenoid, invoking initial conditions and boundary conditions, and performing the numerical analysis. All of these are carried by a series of commands that are read from a COSMOS input file. The input file that corresponds to this example problem is given in Appendix A.

In order to create the two-dimensional axisymmetric geometry of the solenoid, COSMOS requires the input of keypoints that act as endpoints of curves. Curves that make up the geometry of the solenoid can be specified with respect to the keypoints. Figure 2.9 shows the keypoints and curves used to define the separate material regions of the example problem solenoid. Since the solenoid is axisymmetric, the electromagnetic behavior is identical on each side of the center-line. As a result, only half of the two-dimensional cross-section is required to fully analyze the entire solenoid.

After the geometry of the solenoid is completely defined, the separate material regions are meshed with a group of triangular elements (Figure 2.10). Meshing is an automatic feature of COSMOS finite element software. To mesh a region of a model,

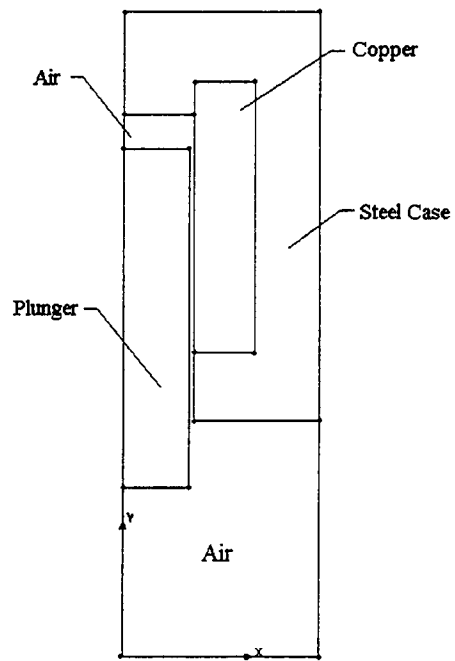


Figure 2.9. Keypoints and Curves of Example Finite Element Problem.

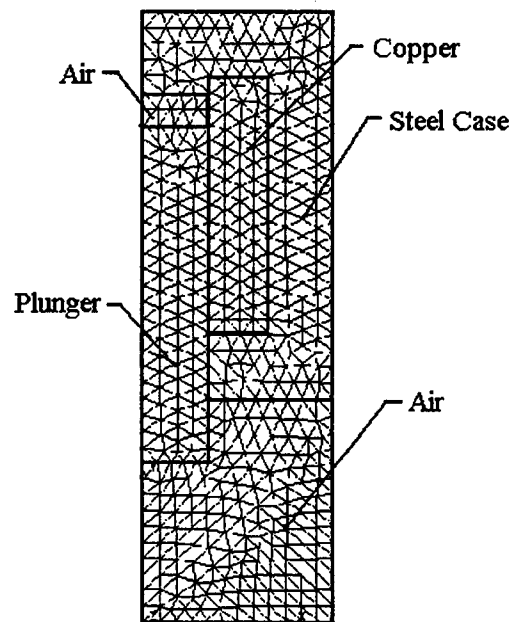


Figure 2.10. Finite Element Mesh of Example Problem Solenoid.

the curves that define the boundaries of that region are specified along with an average element size. Upon creating a mesh for a particular material region, appropriate material properties are specified. As explained in the previous sections, electromagnetic analysis requires specification of the material permeability. To account for the B-H nonlinearity of materials, COSMOS requires the input of ordered pairs that make up the B-H curve for the material. The B-H input data for the verification problem is given in Table 2.2.

After the model is meshed and the material permeabilities are specified, the current flowing through the coil must be specified. At the same time, the number of

Table 2.2. B-H Curve Data Points for COSMOS Verification Problem.

H (W/m²)	B (Ni/m)
460	0.80
640	0.95
720	1.00
890	1.10
1020	1.15
1280	1.25
1900	1.40
3400	1.55
6000	1.65

turns that make up the coil must be defined. Recall that the product of the current, I , and the number of turns, N , is the electromotive force. It is the electromotive force that 'drives' the flux through the magnetic circuit. With COSMOS, the electromotive force of the solenoid is defined by specifying the current density of the region that makes up the coil cross section. The current density of this rectangular area is given by,

$$\text{Current Density} = \frac{I \cdot N}{A_c} \text{ (Amps/m}^2\text{)}, \quad (2.11)$$

where I is the current, N is the number of turns and A_c is the cross-sectional area of the rectangular copper coil region.

After specifying the current density of the copper region, the boundary conditions of the model must be defined. As seen in Figure 2.10, regions of air exist within the finite element model of the solenoid. By calculating the magnetic flux in these free space regions, leakage flux that escapes from the magnetic components of the solenoid is accounted for. In order to specify the absolute boundaries of the system, a line of zero electric potential is defined around all four sides of the finite element model. This is the boundary condition for the electromagnetic solenoid model.

The final step in the finite element analysis process is to carry out the numerical analysis. Once a model is meshed and boundary conditions are supplied, the electromagnetic analysis is carried out. After calculation is complete, the program is able to list and/or plot a number of parameters. These include magnetic fluxes and their Cartesian vector components, as well as magnetic forces. An example of the force output information as displayed in the COSMOS output file is shown in Figure 2.11. The

represents the force output of the solenoid. Since the model is axisymmetric, the force is given in units of Newtons per Radian. The force value must be multiplied by 2π to determine the overall force exerted on the three-dimensional plunger.

```
*****
                        Resulting Forces:
Total X-Force =    5.213
Total Y-Force = -23.542
Total Z-Force =    0.00
*****
COSMOS / M                      Geostar V1.75
```

Figure 2.11. *Example of COSMOS Output File Showing Force on Plunger.*

Once the calculation is completed for the example problem, the answer is compared to the value specified in the COSMOS User's Guide. A matching value means that the methods used in performing the finite element analysis is correct. Only a few modifications are required for the analysis of the small push-pull solenoid. The finite element analysis of the small push-pull solenoid is described in Chapter IV.

CHAPTER III

Outline of Methodology

Introduction

Given the background information that has been presented, the focus of this report returns to the task of producing an optimum solenoid design. The purpose of this chapter is to outline the methodology used in this optimization study. In doing so, it will be shown how finite element output along with solenoid cost information is used to produce a cost-optimal design. This chapter only briefly describes the optimization methodology while the individual steps are explained in more detail in analysis chapters of this report.

As discussed in Chapter II, solenoid performance is evaluated as the force delivered by the armature over its stroke distance. Solenoid design specifications are most commonly defined by a minimum amount of force delivered by the armature at a specified stroke position. For example, a small push-pull solenoid may be specified to require a force of 12 lb. delivered at a stroke position of 0.1 inches. In order to conform to such a requirement, the goal of the final algorithm produced by this study is to allow for the specification of minimum force at a given stroke distance. The end result will be the definition of the solenoid design that meets these specified requirements at the lowest cost.

Design Variables

The first objective of this study is to describe solenoid performance with respect to a set of design variables. In order to do so, the factors that effect solenoid performance must be identified. As discussed in Chapter II, these factors may include the geometric configuration of the solenoid as well as the properties of materials of which it is made. The design variables that will be analyzed in this study are shown in Figure 3.1. These variables include solenoid length, L ; outside case radius, C_R ; plunger radius, P_R ; wall thickness, W ; notch depth, D ; notch angle, A ; and stroke position S . While other parameters could be included, these variables were selected because they most greatly effect the force output of the solenoid. The remaining dimensions for the solenoid are held constant throughout the study and are presented in the *Assumptions* section of this chapter.

As these design variables are changed, the electromagnetic circuit that describes the physics of the solenoid is altered. As a result, each of an infinite range of possible configurations produces a unique force output. The effect of each design variable on force output may be characterized by using the finite element method. By creating and analyzing finite element models over a range of values for a particular design variable, its effect on force output can be observed. An algebraic expression of force as a function of a particular design variable may then be achieved by interpolating a mathematical function to fit the data obtained. Analysis of each design variable in this way produces an expression of force as a function of all design variables. Upon developing an

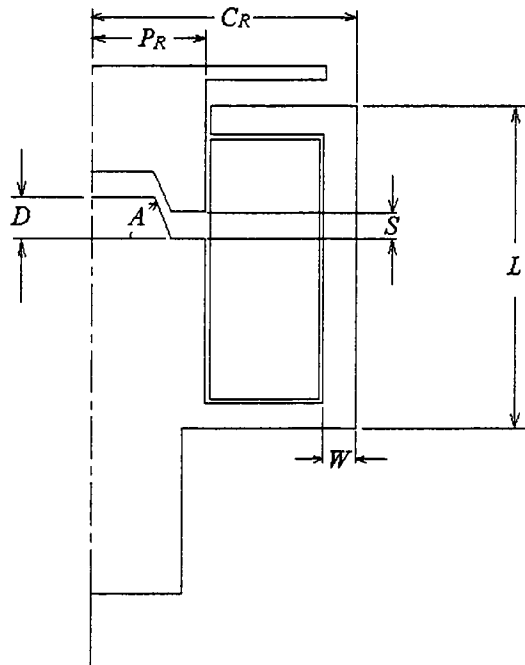


Figure 3.1 Design Variables of the Solenoid.

algebraic expression of cost with respect to the same set of design variables, minimization techniques are used to identify the cost-optimal solenoid.

Two Phases of Study

This study has been divided into two phases of analysis. The first phase involves the analysis and optimization of the pole interface configuration. Recall that the pole interface is found in the center of the solenoid where the plunger interfaces with the inner steel core. The pole interface is characterized by the notch depth, D , and notch angle, A ,

which are shown in Figure 3.1. The remaining variables shown in Figure 3.1 are analyzed and optimized in the second phase of this study.

There are three main reasons that this optimization study is divided into two separate phases. The first reason is that the design variables that describe the pole interface, unlike the phase two design variables, do not effect the cost of the solenoid. The notch depth and notch angle simply describe the shape of the interface and hence do not contribute to the amount a material (steel) required to produce the solenoid. Since these variables do not effect cost, they are analyzed separately from those variables that require cost minimization.

The second reason for having two separate analysis phases involves the finite element models used in each phase. Since phase one deals only with the effects of the pole interface, a special finite element model is used in order to decrease the reluctance effects within the remaining components of the solenoid. This involves increasing the thickness of the solenoid material within the finite element model to isolate the effect of the pole interface. This idea is described in detail in the following chapter.

The third reason for having two separate phases is that it decreases the amount of computation necessary to perform the least squares interpolation. As will be described in chapter five, the least squares method of interpolation requires a substantial amount of computation. The amount of computation required to fit a polynomial to a set of data points increases exponentially with the number of independent variables. By having two separate equations of force, the necessary computation is much less than would be required for a single force equation.

Phase One

As stated, the goal of phase one is to identify the pole configuration that produces a maximum force at a specified stroke distance. The development and analysis of finite element models produce a number of data points that are used to characterize the effect of each design variable. In phase one, each data point consists of a notch depth, notch angle, stroke position and the corresponding force value. A least squares interpolation is performed on the set of data points to produce an empirical formula that describes force as a function of the notch depth, D , notch angle, A , and stroke position, S . This function is written in general form as,

$$force = f(D, A, S) \quad (3.1)$$

By supplying a stroke position to this equation, the notch depth and notch angle that produce the maximum force can be found.

Phase Two

The second phase of analysis is similar to the first aside from the addition of a two design variables. By producing a matrix of data points by finite element analysis, a second interpolation function is determined. This equation describes force as a function of the remaining cost-dependent design variables and has the general form,

$$force = f(L, C_R, P_R, W, S) \quad (3.2)$$

In addition to the force function, phase two requires the development an expression of cost as a function of the same variable set. Realizing that cost is independent of stroke position this function would have the general form,

$$Cost = f(L, P_R, C_R, W) \quad (3.3)$$

By specifying a minimum force requirement at a specific stroke position as a constraint, the cost function can be minimized to define the cost-optimal design configuration.

Assumptions

As with any engineering problem, a clear statement of all assumptions is critical. As previously mentioned, all geometric dimensions that are not included as a design variable are held constant. Figure 3.2 shows a diagram of the solenoid with all necessary dimensions given in inches. The appropriate symbol appears for dimensions corresponding to design variables.

The following is a list of additional assumptions that apply to this study.

- Constant current of 3.2 amps
- Steady-state system that neglects thermal effects on solenoid performance
- Magnetic permeability is the same for all steel components and corresponds to that of the experimental solenoid (see Chapter IV)
- Relative magnetic permeability of brass and nylon bobbin equal to one (same as air)

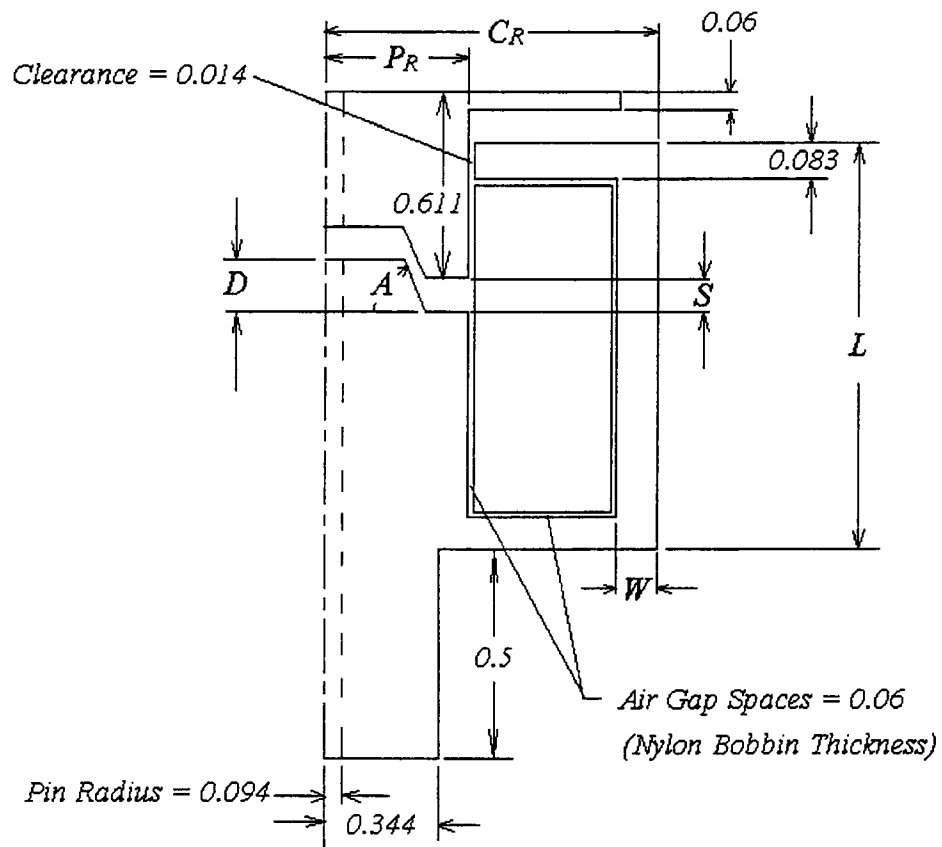


Figure 3.2. Assumed Dimensions and Design Variables.

Methodology Summary

The flow diagram given in Figure 3.3 summarizes the various steps that are followed in this study. After a force and stroke position are given, the next step is to determine the optimal pole configuration. The configuration defined here is optimal at that particular stroke position. Next, the force and stroke position are supplied to the

phase two force function. The cost function is then minimized with the minimum force constraint equation. The result is the solenoid that produces the largest possible force at a particular stroke position for the lowest possible cost. The following chapters give detailed descriptions of the steps shown in Figure 3.3.

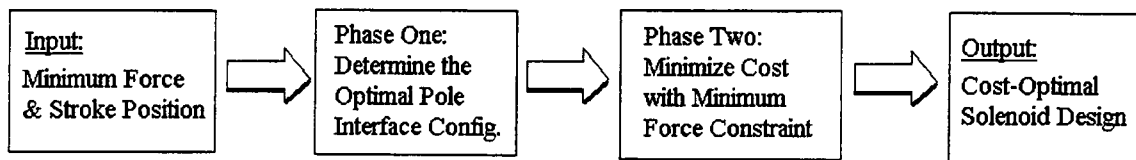


Figure 3.3. Flow Diagram Showing Steps Used to Produce a Cost-Optimal Solenoid.

CHAPTER IV

Finite Element Analysis of the Solenoid

Introduction

As described in the Chapter II, COSMOS finite element software is used in this study to perform the necessary numerical analysis that describes the behavior of the solenoid. This numerical analysis allows for the calculation of the force exerted by the armature of the solenoid. The finite element analysis process is described in the form of an example problem in chapter two. Few modifications are required in modeling the small push-pull solenoid. The purpose of this chapter is to illustrate the process by which finite element models are used to create a foundation of data to be used later in the development of an algebraic force function.

The initial step in the analysis process is to create a finite element model of the small push-pull solenoid. Part drawings for the actual solenoid are used to generate the geometry of the finite element model. Along with geometric details, information about the coil and electric current are required to fully describe the model. For example, for this particular small push-pull solenoid, the number of turns within the copper coil is 506. The electric current used in experimental testing of the specimen was 3.2 amps. By knowing the cross-sectional area of the coil, (2.11) is used to calculate the current density of the coil region. As with the example problem, the solenoid model is meshed and

material properties are defined. The B-H curve data from the verification problem is initially used to define the magnetic behavior of the steel. Once the modeling process is complete, numerical computation is carried out to evaluate the force on the plunger. The COSMOS input file used in analyzing the small push-pull solenoid is given in Appendix B. Figure 4.1 shows the finite element model for the small push-pull solenoid.

Solenoid performance is usually evaluated by examining the force output of the actuator over the stroke distance of the plunger. For the small push-pull solenoid, the stroke distance is approximately 0.2 inches. In order to evaluate the solenoid force

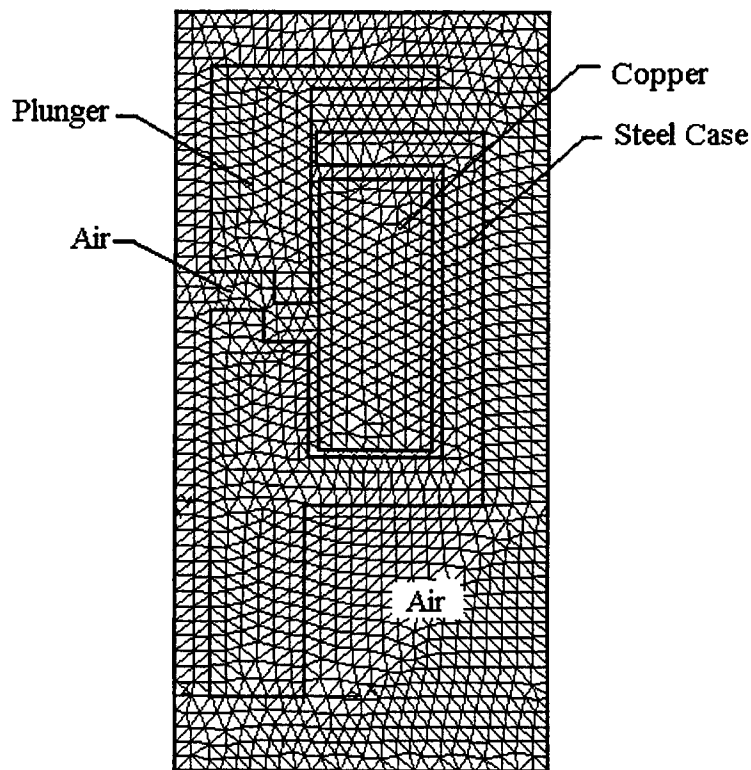


Figure 4.1. *Finite Element Model of the Small Push-Pull Solenoid.*

output over an entire stroke, a number of finite element models are required at various positions along the stroke. By translating the keypoints that define the plunger geometry, a new finite element model is easily generated. In this way, all input files for the finite element model are identical aside from a keypoint relocation statements that describes the plunger configuration with respect to a datum. By analyzing each of these models, the force over the entire stroke distance is calculated. The finite element data is then compared to experimental test data in order to assess the accuracy of the finite element method.

Figure 4.2 shows a plot of the experimental test data for the small push-pull solenoid [16]. As seen in the figure, the force is nonlinear with respect to stroke position, which is typical of most solenoid designs. For this reason, at least three models are required to estimate the path of this force function. For the purpose of comparing finite element results with experimental data, the finite element analysis is performed at 0.02 inch increments throughout the stroke distance. Figure 4.3 shows the finite element analysis results along with the experimental data. The actual experimental and finite element data is given in Table 4.1.

While Figure 4.3 shows some correlation between experimental and finite element data, the two curves begin to diverge at small stroke positions. Recall that the B-H curve data from the example problem was used in these finite element models. Since the B-H curve provides a correlation between magnetic intensity and flux density in the steel of the solenoid, it is undoubtedly a crucial part of the model description. The magnetic behavior of steel depends heavily on chemical composition and machining

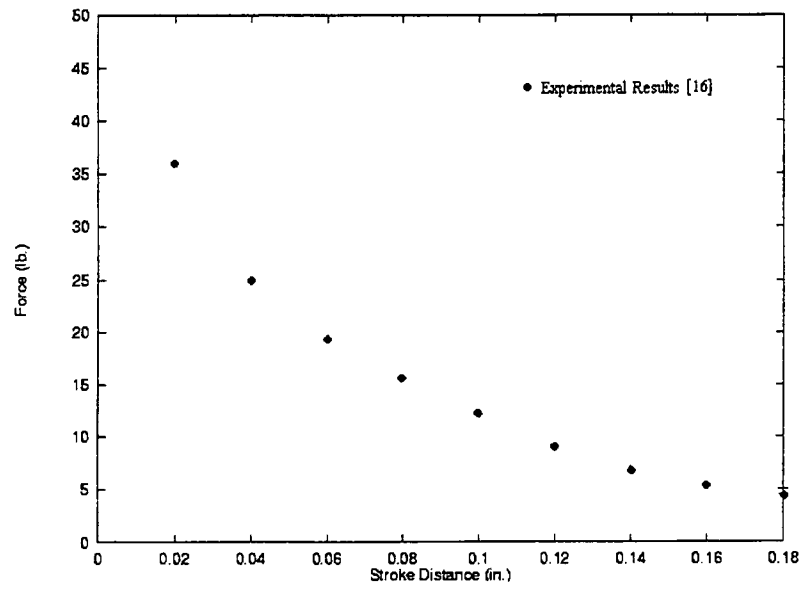


Figure 4.2. *Experimental Test Data Showing Force vs. Stroke Position.*

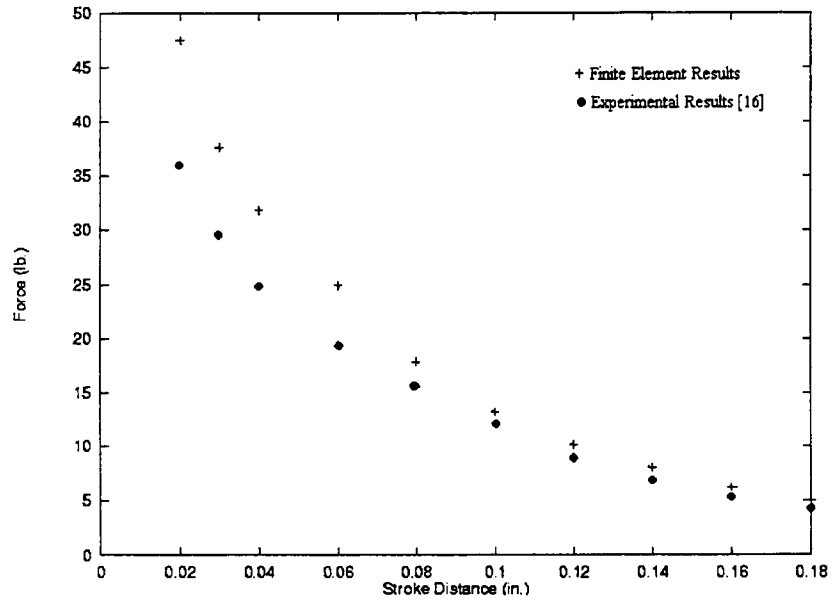


Figure 4.3. *Experimental and Finite Element Results with Modified B-H Curve*

Table 4.1. Finite Element and Experimental Results.

	Finite Element	Experimental
Stroke (in.)	Force (lb.)	Force (lb.)
0.02	47.494	35.956
0.04	31.837	24.886
0.06	24.900	19.392
0.08	17.845	15.675
0.10	13.272	12.201
0.12	10.155	9.050
0.14	8.019	6.868
0.16	6.251	5.414
0.18	5.029	4.444
0.20	3.889	3.959

processes and as a result, is difficult to exactly predict without experimentation [13]. To account for this problem, B-H curves for different types of steel were compared and data points were found that produce results that more closely correspond with the empirical data. These modified B-H data points are given Table 4.2. Figure 4.4 shows the B-H curves used in the example problem along with the modified B-H curve. While these curves exhibit a subtle difference, the effect on the numerical result is quite significant. Figure 4.5 shows the force profiles for the experimental data and force data from the newly modified finite element model. As shown in the figure, the modified B-H curve produces forces that closely correspond with the experimental results. The experimental and finite element results obtained with the modified B-H curve is given in Table 4.3.

Table 4.2. *Modified B-H Data Points.*

B (T)	0.66	0.79	0.83	0.91	0.95	1.04	1.16	1.29	1.37
H (A/m)	410	590	670	840	970	1230	1850	3350	5950

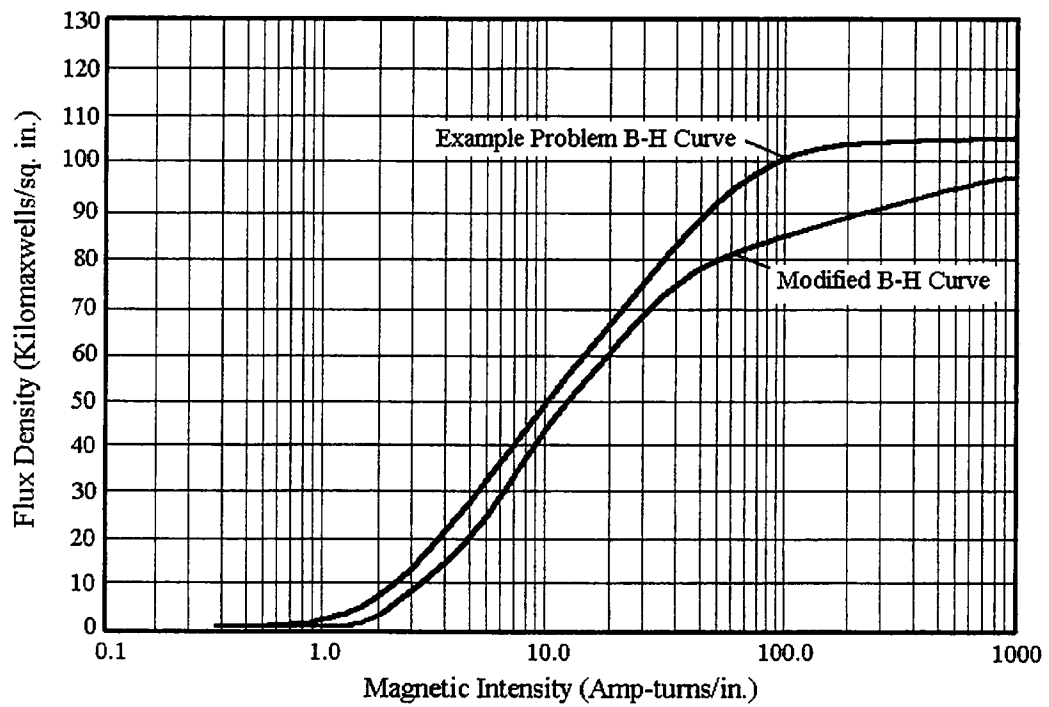


Figure 4.4. *Example Problem and Modified B-H Curves.*

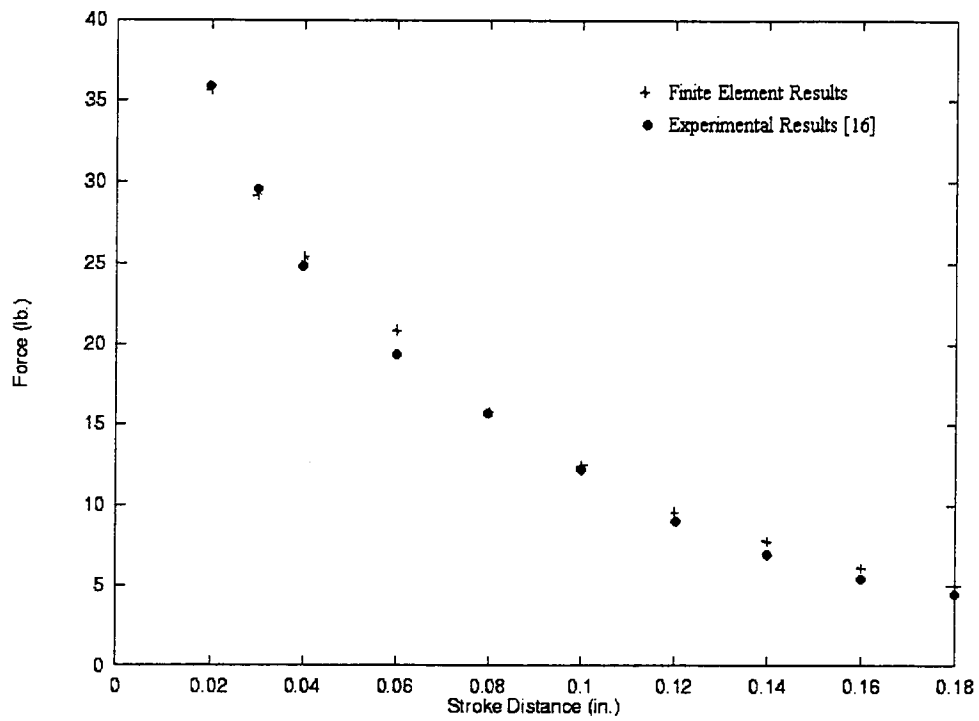


Figure 4.5. *Experimental and Finite Element Force Plots From Modified B-H Curve.*

Table 4.3. *Finite Element and Experimental Results From Modified B-H Curve.*

	Finite Element	Experimental
Stroke (in.)	Force (lb.)	Force (lb.)
0.02	35.645	35.956
0.03	29.152	29.492
0.04	25.388	24.886
0.06	20.857	19.392
0.08	15.796	15.675
0.10	12.207	12.201
0.12	9.606	9.050
0.14	7.728	6.868
0.16	6.102	5.414
0.18	4.953	4.444

Convergence Study

A second step in evaluating the accuracy of the finite element analysis is to perform a convergence study. A convergence study provides a means for assessing the quality of the finite element method and is independent of the type of problem being solved. The study also lends useful information about the type of mesh that should be used to obtain a sufficiently accurate answer.

To perform a convergence study on the finite element model of the solenoid, a number of different models are generated with varying mesh densities or element sizes. Models with relatively large elements are referred to as having a coarse mesh. A mesh made up of many small elements is referred to as a fine mesh. By analyzing the finite element model with a range of meshes from coarse to fine, the results should converge to a particular value. If this occurs, then the model is said to be convergent.

The mesh density of the finite element model is controlled by specifying the average element size. In this way, a smaller average size will produce a more refined mesh. Models are analyzed with average element sizes of 0.001 m, 0.0009 m, 0.0008 m, 0.0007 m, and 0.0006 m. These numbers refer to the average length in meters of a single side of a triangular element. Figure 4.6 shows the calculated force on the plunger for each of these four models. It can be seen from the figure that the model is convergent.

As stated, a convergence plot such as the one shown in Figure 4.6 may lend useful information about the mesh density that should be used in the finite element models. As shown in the figure, there is little difference between the solutions produced by the three

models having the most elements. This may lead to the belief that the third mesh size (0.0008 m) is optimal since it produces the most accurate result with a tolerable amount of computation. In most analysis problems this would be true. However, the analysis being performed in this optimization study is mostly concerned with relative changes in force output caused by changes in solenoid geometry. As a result, a coarse mesh size (0.001 m) can be used and much computational time is eliminated. A coarse mesh is especially desirable in this situation that involves the analysis of over a hundred finite element models and only a small amount of accuracy is lost.

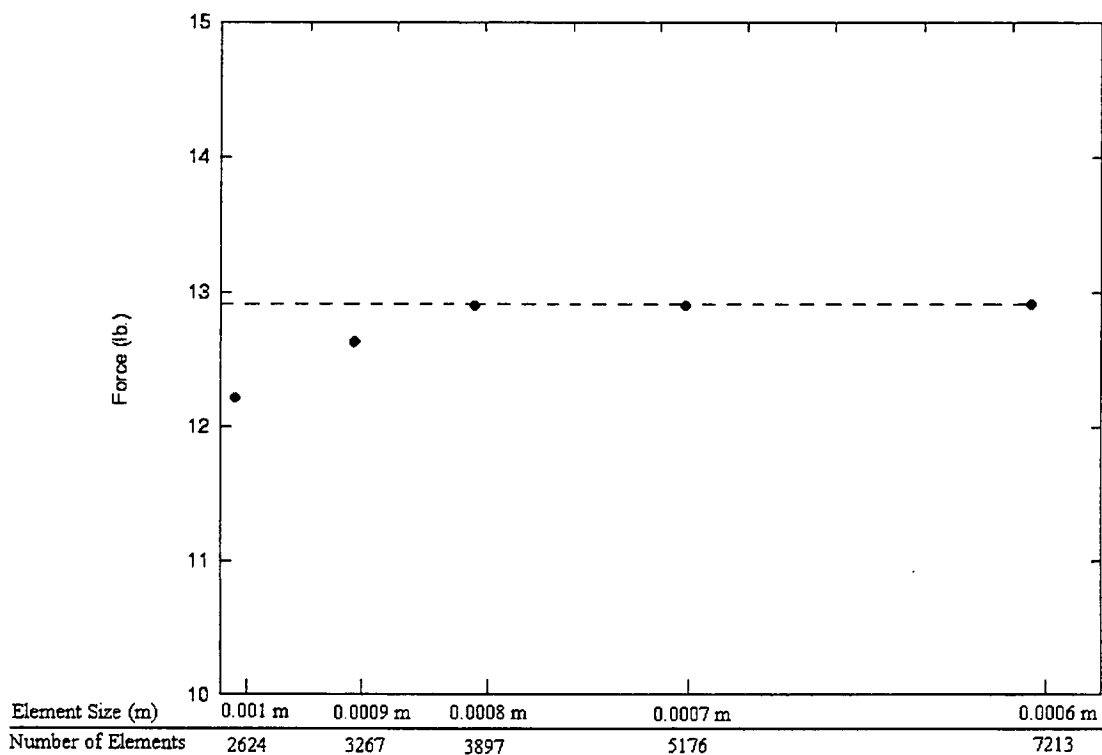


Figure 4.6. Convergence Study Results.

Creating a Database

As described in Chapter III, the finite element method is used to characterize the effect of each design variable on the performance of the solenoid. For example, performing finite element analysis on solenoid models with varying outside diameters shows the relation between solenoid performance and diameter. By using finite element analysis in this way on each design variable, a database is created that describes the effect of each design variable on solenoid performance. Much like a parametric study, this is achieved by creating and analyzing models at varying geometric increments. Once each design variable is analyzed over a predetermined geometric range, the result is a large matrix of data that is used to develop an algebraic expression of force as a function of each design variable.

To increase the practicality of the methods presented in this study, the number of finite element models used to create the necessary database should be minimized. As stated previously, the estimation of the nonlinear relationship between force and stroke distance requires at least three data points. Similarly, a nonlinear relationship between force and a given design variable may require at least three models. The number of finite element models that are required also depends on the geometric range over which the design variables are varied. For example, if the solenoid diameter is varied and analyzed over a relatively large range, a larger number of models should be used.

Phase One Analysis

Phase one of the analysis process as described in Chapter III is to determine the effects of the pole interface on solenoid performance. Recall that the pole interface is at the center of the solenoid where the plunger meets the inner core. The design variables analyzed in phase one are notch depth, D and notch angle, A (Figure 4.7). A conical configuration as shown in Figure 4.8a is most effective in producing large forces at stroke distances greater than 0.06 inches. For shorter stroke distances, the flat-faced configuration shown in Figure 4.8b is most effective [11]. The purpose of the phase one parametric analysis is to fully describe the effects of pole interface geometry and stroke position on solenoid performance. In order to do so, the pole interface design variables are analyzed at a number of stroke positions.

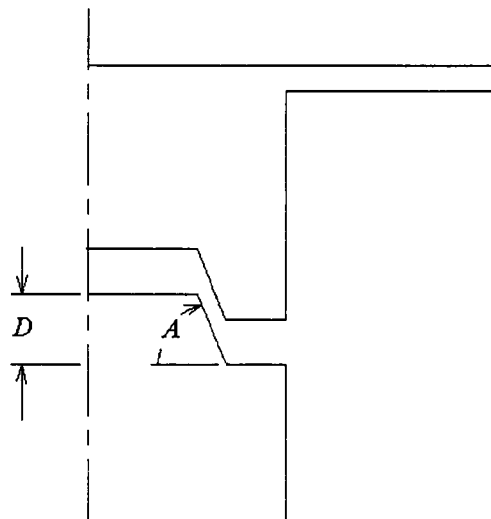


Figure 4.7. Phase One Design Variables.

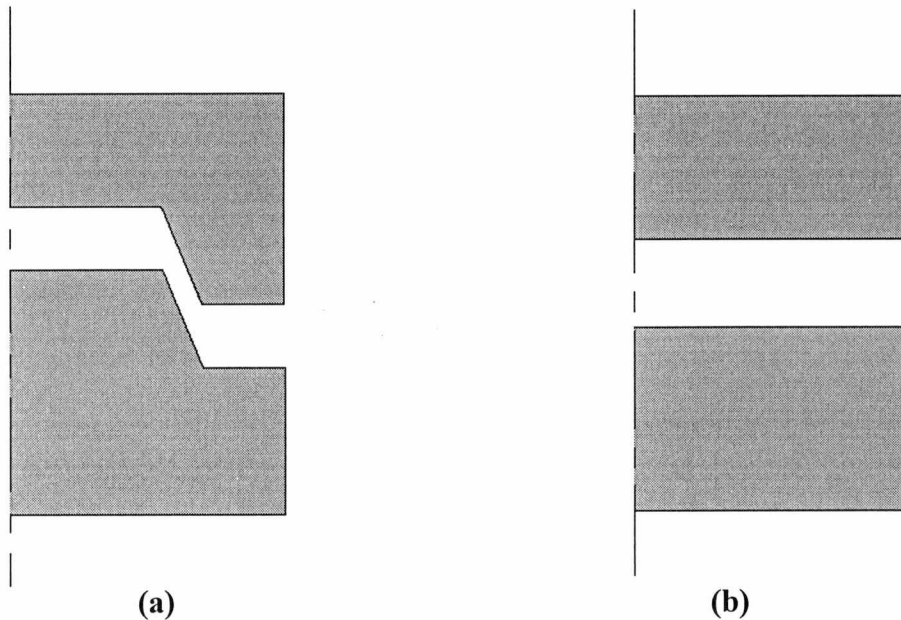


Figure 4.8. *Conical Pole Interface Configuration(a) and Flat-Faced Pole Interface Configuration(b).*

In the phase one analysis, for each increment of a design variable, five separate models are used to describe the force over the stroke distance. These stroke positions are 0.4, 0.6, 0.8, 0.11, and 0.14 inches. The next step is to choose the range over which each design variable is analyzed. It has been documented that a 30° notch angle produces an optimum force for larger stroke distances. The 90° notch angle, when combined with a notch depth greater than one, describes a step-faced configuration. When a notch angle is combined with a notch depth equal to zero, the result is a flat-faced configuration (Figure

4.8b) which has been documented to produce larger forces at smaller stroke positions [11]. To capture the full range of geometric possibilities, the notch angle is analyzed between 25° and 110° (Note: 110° notch angle is not a valid design consideration since this angle would not allow the plunger and core to mate). Therefore, analysis at all five stroke positions is performed for angles of 25° , 30° , 45° , 60° , 75° , 90° and 110° . The range of study for the notch depth is chosen to be from 0.02 and 0.14 inches at increments of 0.02 inches. Again, each configuration is analyzed at all five stroke positions to best describe their interdependence. The geometric range of each design variable is summarized in Table 4.4.

As mentioned in Chapter III, a special type of finite element model is used to analyze the pole interface configuration. The purpose of this special model is to reduce the reluctance effects within the solenoid thereby isolating the pole interface as the main

Table 4.4. *Geometric Range of Notch Depth and Notch Angle.*

D (in.)	A
0.02	25°
0.04	30°
0.06	45°
0.08	60°
0.10	75°
0.12	90°
0.14	110°

source of reluctance. In this way, the pole interface configuration may be analyzed and optimized independently. The low reluctance model is achieved by increasing the thickness of the outer walls. This simulates the effect of increasing area of the cross-section through which the magnetic flux travels. Figure 4.9 Shows the geometry of the low reluctance model.

Phase one analysis requires the development and analysis of 65 finite element models. An example of the results obtained from phase one analysis is given in Table 4.5. The entire data set is given in Appendix D of this report. Recall that the

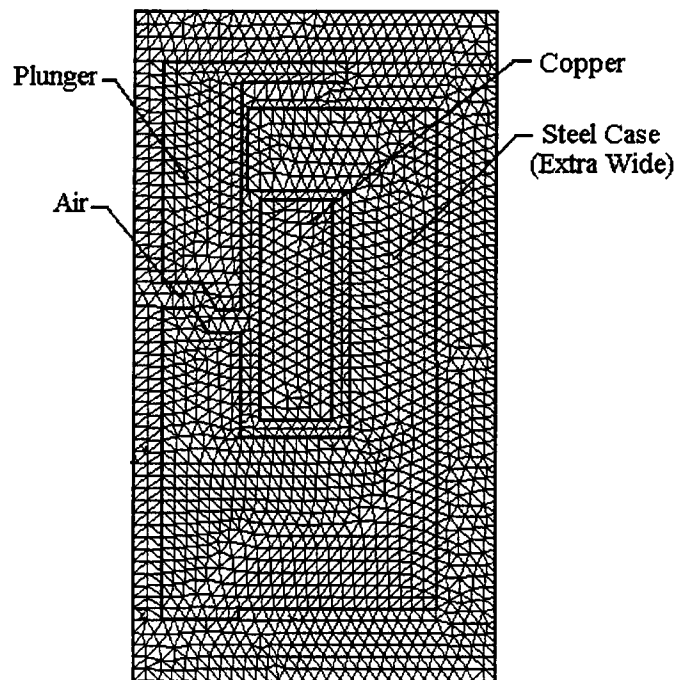


Figure 4.9. *Low Reluctance Finite Element Model
Used for Phase One Analysis.*

Table 4.5. Example Phase One Analysis Results.

S (in.)	D (in.)	A	Force (lb.)
0.04	0.08	60°	29.532
0.06	0.08	60°	20.394
0.08	0.08	60°	14.083
0.11	0.08	60°	8.635
0.14	0.08	60°	5.872

purpose of the data produced in the phase one is to ultimately produce an algebraic expression of force as a function of stroke distance, notch depth, and notch angle. This function is then used to predict the optimal pole configuration for a specified stroke distance. The creation of this algebraic expression and the optimization techniques are described in later chapters.

Phase Two Analysis

Phase two includes the analysis of those design variables that effect the material cost of the solenoid. The design variables that are analyzed in phase two are shown in Figure 4.10. They are solenoid length, L ; outside case radius, C_R ; plunger radius, P_R ; and the outside wall thickness, W . As in phase one, these parameters must be varied and analyzed over some predetermined range and at a number of stroke positions. In phase

two, analysis is performed at stroke positions of 0.04, 0.06, 0.08, and 0.11 inches. The ranges over which the variables are analyzed are summarized in Table 4.6. These ranges were chosen in order to encompass the optimal configuration for each design variable.

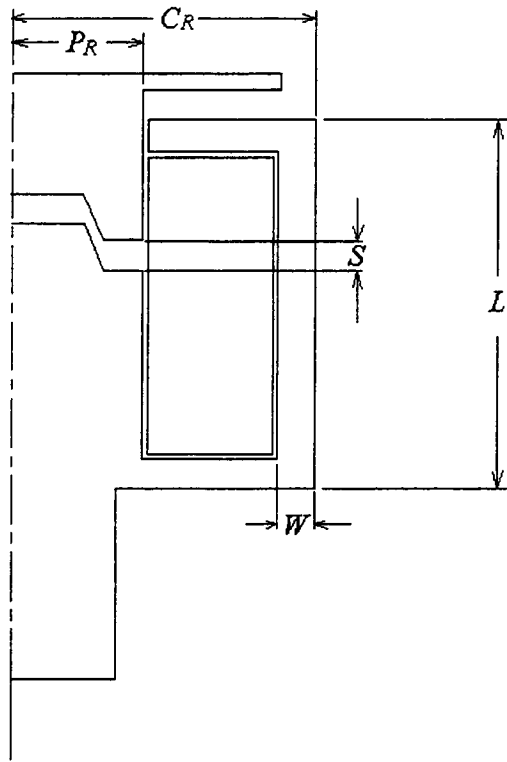


Figure 4.10. *Phase Two Design Variables.*

Phase two analysis requires the development and analysis of 104 finite element models. An example of the data obtained from the analysis is given in Table 4.7 while the entire data set is given in Appendix E. As in phase one, this data set is used to create

Table 4.6. *Geometric Ranges of Phase Two Design Variables.*

L	C_R	P_R	W
0.673	0.691	0.298	0.074
0.770	0.731	0.315	0.084
0.866	0.772	0.333	0.095
0.962	0.813	0.350	0.105
1.058	0.853	0.368	0.116
1.154	0.894	0.385	0.126
1.251	0.934	0.403	0.137

an algebraic function of force with respect to the four geometric variables as well as stroke position. It will be shown that the development of algebraic expressions for the phase one and phase two sections of this study are nearly identical. Unlike phase one, however, information about solenoid cost shall be incorporated into phase two in order to predict the cost-optimal solenoid configuration.

Table 4.7. *Example of Phase Two Analysis Results.*

S (in.)	L (in.)	C_R (in.)	P_R (in.)	W (in.)	Force (lb.)
0.04	0.962	0.813	0.35	0.105	25.388
0.06	0.962	0.813	0.35	0.105	20.856
0.08	0.962	0.813	0.35	0.105	15.795
0.11	0.962	0.813	0.35	0.105	10.988

CHAPTER V

Method of Least Squares

Introduction

The least squares method is used here to curve fit the data from phases one and two to produce two separate polynomials. The first of these algebraic expressions will use the data from the phase one analysis to describe force as a function of the pole interface design variables and stroke position and will have the general form,

$$force = f(S, D, A) \quad (5.1)$$

where S is stroke position, D is notch depth, and A is notch angle.

The second expression will be developed from the phase two data and will express force as a function of stroke position, S , and the remaining cost-dependent design variables. It will have the general form,

$$force = f(S, L, C_R, P_R, W) \quad (5.2)$$

where L is the length of the solenoid, C_R is the radius of the outside casing, P_R is plunger diameter, and W is the outside wall thickness of the solenoid (Figure 3.1). The remainder of this chapter discusses the theory of the least squares method for curve fitting and how it will be used to develop the expressions in (5.1) and (5.2).

Simplified Example

To explain the least squares method and how it is used to curve fit a polynomial to a given data set, a simple example is considered. Suppose that a data set is given in the form of two vectors, X and Y , below.

$$X = [0 \ 0.3 \ 0.8 \ 1.1 \ 1.6 \ 2.3]$$

$$Y = [0.5 \ 0.82 \ 1.14 \ 1.25 \ 1.35 \ 1.40]$$

A plot of this data is given in Figure 5.1.

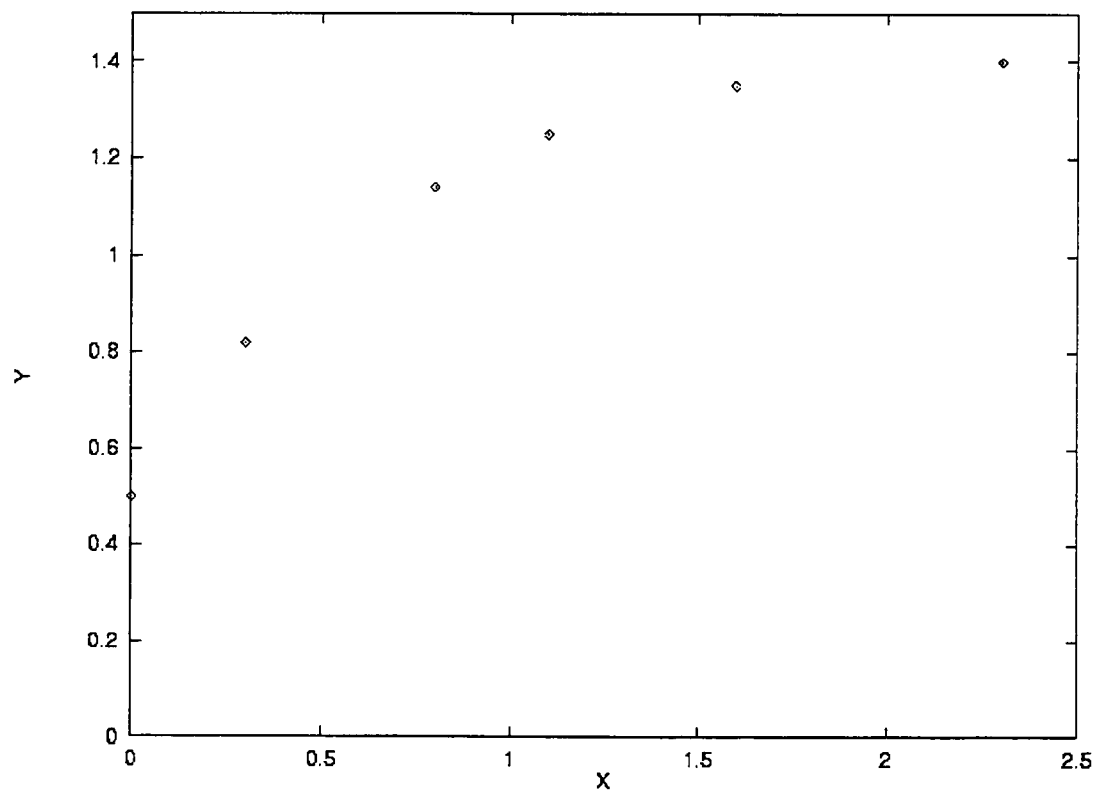


Figure 5.1. Example Data Set.

Ideally, the desire is to find a smooth function $y=f(x)$ that closely fits the data points. Since y is clearly nonlinear with respect to x , at least a second order polynomial is required to closely approximate a curve through the data set. The quadratic expression of force with respect to x would have the form,

$$y = a + bx + cx^2 \quad (5.1)$$

The goal of the least squares method is to determine the values of the three constants a , b , and c , such that the resulting algebraic expression best approximates the data set.

The fundamental principal of the method of least squares is to minimize the square of the error, η , given by the relation:

$$\eta_i = y_i - (a + bx_i + cx_i^2) \quad (i = 1, 2, \dots, 6) \quad (5.2)$$

The method then consists of determining of a , b , and c in such a way that,

$$Q = \sum_{i=1}^6 \eta_i^2 \quad (5.3)$$

is minimized. To find the values of a , b , and c , the following equations are formed.

$$\frac{\partial Q}{\partial a} = -2 \sum_{i=1}^6 (y_i - a - bx_i - cx_i^2) = 0 \quad (5.4)$$

$$\frac{\partial Q}{\partial b} = -2 \sum_{i=1}^6 x_i (y_i - a - bx_i - cx_i^2) = 0 \quad (5.5)$$

$$\frac{\partial Q}{\partial c} = -2 \sum_{i=1}^6 x_i^2 (y_i - a - bx_i - cx_i^2) = 0 \quad (5.6)$$

These three equations are solved simultaneously to find the values of a , b , and c which gives the following quadratic equation,

$$y = 0.5318 + 0.9191x - 0.2387x^2 \quad (5.7)$$

When this expression is plotted along with the original data set, it can be seen how well the method of least squares fits a function to the data (Figure 5.2).

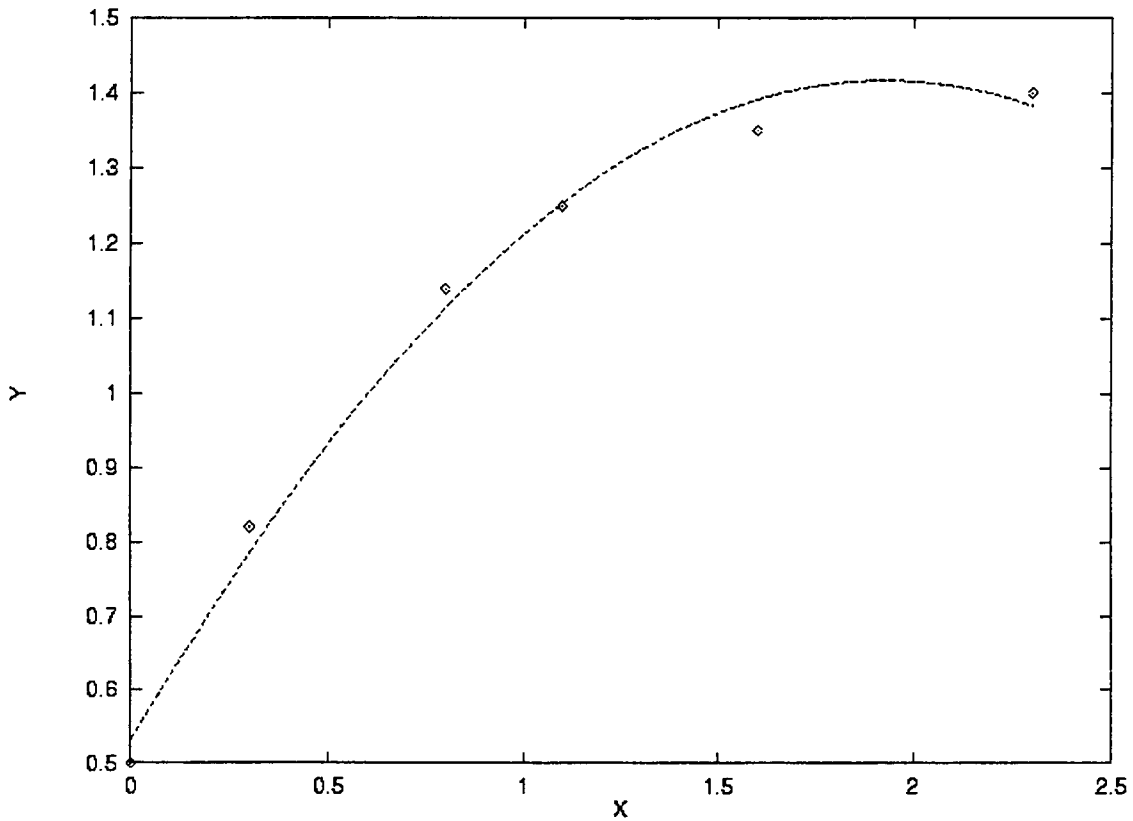


Figure 5.2. Data Set with Quadratic Curve Fitting Function.

In order to increase the accuracy of the interpolating function, a higher order polynomial may be used. For example, a third degree polynomial of the form,

$$y = a + bx + cx^2 + dx^3 \quad (5.8)$$

may produce a curve that comes closer to passing through each data point. To produce a third order polynomial, (5.1) is again differentiated with respect to each of the unknown coefficients a , b , c , and d . By solving for these coefficients in the same way as before, the following third degree polynomial is obtained.

$$y = 0.50 + 1.20x + 0.57x^2 + 0.10x^3 \quad (5.9)$$

When plotted along with the data set (Figure 5.3), there is a noticeable improvement in accuracy as compared to the quadratic approximation.

Further, Figure 5.4 shows a fourth order interpolation function which only slightly differs in accuracy from the third order approximation.

These plots lead to an important observation regarding the method of least squares. The solution seems to become increasingly accurate and converge as the degree of the curve fitting function is increased. However, as the number of coefficients approach the size of the data arrays, the polynomial can become unstable and produce oscillations between data points. In fact, the least squares method produces invalid results when the number of unknown constants within the curve fitting function exceeds the number of data points. Figure 5.5 shows how a twelfth order polynomial produces an invalid function with unwanted oscillation.

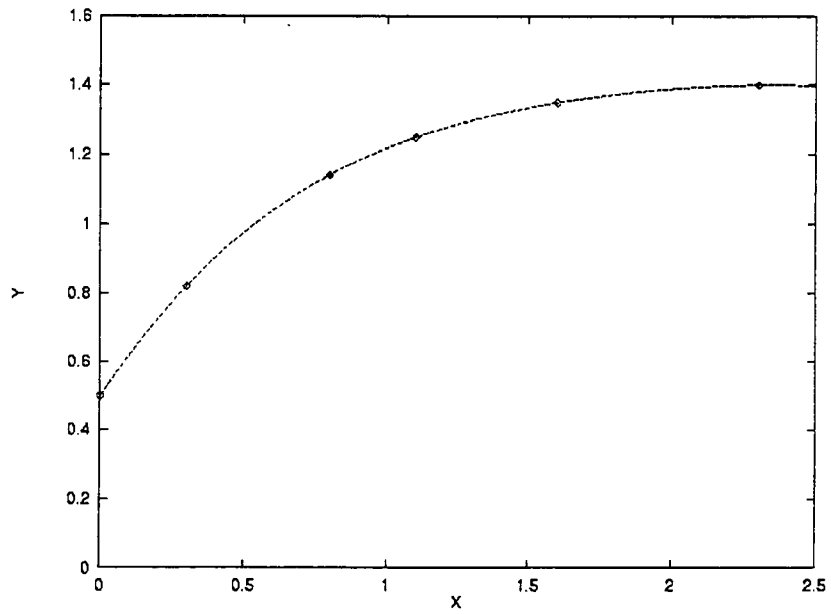


Figure 5.3. Plot of Data Set with 3RD Degree Curve Fitting Function.

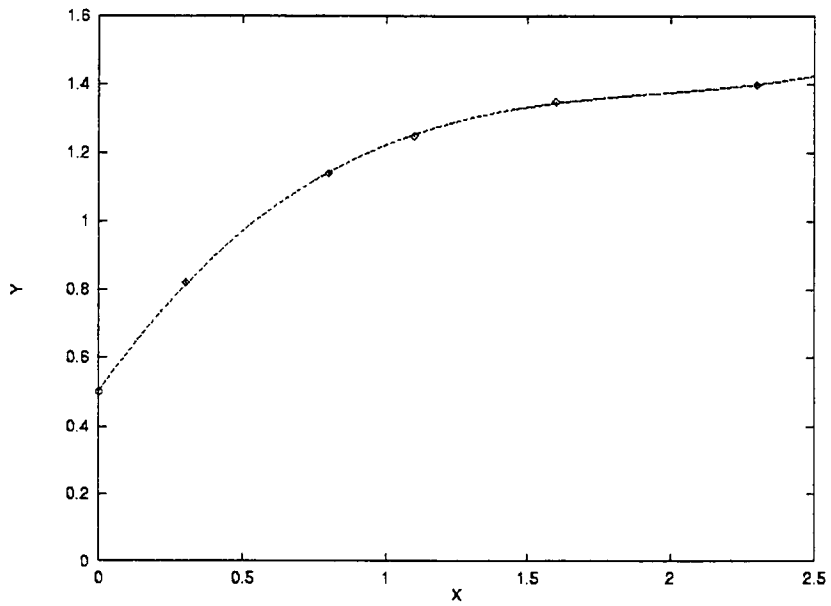


Figure 5.4. Plot of Data Set with 4TH Degree Curve Fitting Function.

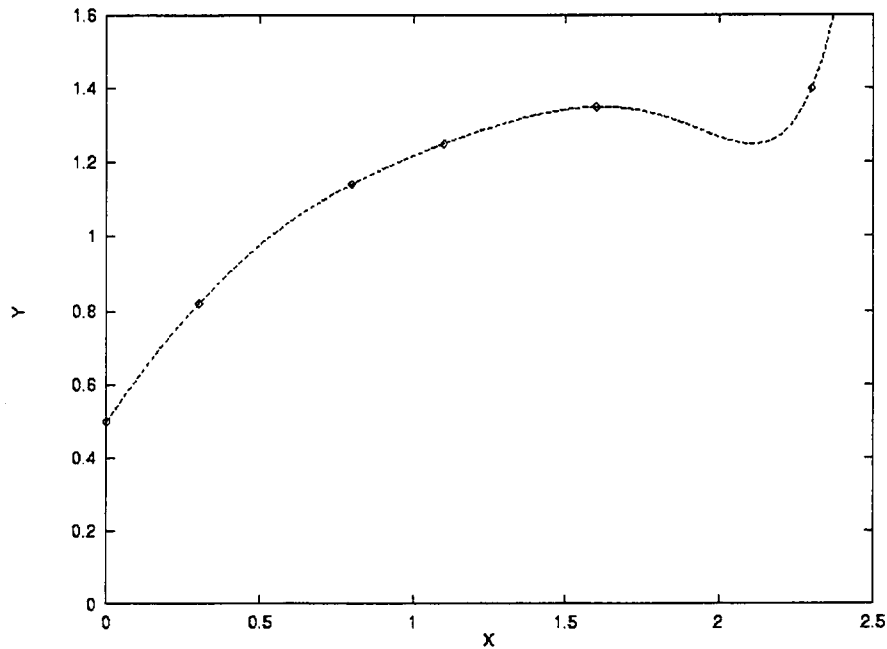


Figure 5.5. *Plot of Data Set with 12TH Degree Curve Fitting Function.*

It is then evident that there exists an ideal polynomial that should be used to curve fit a data set. To reduce computational effort, the ideal interpolating function is the lowest degree polynomial that produces sufficiently accurate results. For this simple example, one could argue that a second degree polynomial is ideal since it almost exactly fits the path of the data set. At the same time, there is a noticeable improvement in accuracy achieved by using a third degree function.

Unfortunately, visual inspection requires the determination and comparison of multiple interpolating functions and is thus impractical. Because of the inability to plot in four dimensions, this method of comparison is especially impractical for interpolating functions having three or more independent variables. For these reasons, an alternative method for predicting the accuracy of curve fitting functions is desired.

The Coefficient of Determination

In order to determine how well the estimated interpolation curve fits the data, we use the following important relationship. If the second degree least squares function is given by,

$$\hat{y} = a + bx + cx^2 \quad (5.10)$$

it can be shown that [4],

$$\sum_{i=1}^6 (y_i - \bar{y})^2 = \sum_{i=1}^6 (y_i - \hat{y}_i)^2 + \sum_{i=1}^6 (\hat{y}_i - \bar{y})^2 \quad (5.11)$$

The quantity given by $\sum (y_i - \bar{y})^2$ measures the sum of squared deviations of the observed y values from $\bar{y}$. This is often referred to as the total corrected sum of squares of y or the ‘total variation’ in y . The quantity $\sum (y_i - \hat{y}_i)^2$ is the residual sum of squares and is the fundamental equation minimized in (5.3). It is sometimes called the ‘unexplained variation’. The quantity $\sum (\hat{y}_i - \bar{y})^2$ represents the variation of the points $\hat{y}_i$ on the estimated interpolation curve and is often called the ‘explained variation’. The important point is that the total variation can be partitioned into two components, the explained and unexplained variation is depicted in (5.12) [4].

$$\text{Total Variation} = \text{Explained Variation} + \text{Unexplained Variation} \quad (5.12)$$

The ratio of the explained variation to the total variation measures how well the curve fits the data set. This ratio is called the coefficient of determination, r^2 and is given by the formula,

$$r^2 = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} \quad (5.13)$$

The coefficient of determination must lie between zero and one. If it is equal to one then all observed data points lie exactly on the fitted curve. If it is equal to zero, then $\hat{y}_i = \bar{y}$ for all i , so the interpolation function is a straight line with a slope of zero. Therefore the closer r^2 is to one, the closer the interpolation function comes to passing through the data points. The coefficient of determination is most effective in evaluating the accuracy of curve fits to large sets of data such as the finite element results considered in this study [4].

Table 5.1 gives the r^2 values for various order polynomial curve fits for this example. Notice the r^2 coefficient only slightly increases as the polynomial degree is increased from two to four. This indicates that a second degree polynomial may in fact be the ideal curve fitting function for this data set. Also notice that the unstable curve fit produced by the twelfth order polynomial approximation produces good r^2 value. This means that another method for identifying function instability (oscillations) must be employed. Later chapters discuss how plots along with random finite element testing are effective in identifying such instabilities. Meanwhile, the r^2 coefficient will prove useful in assessing the quality of the curve fitting functions that describe solenoid force output.

While the calculations necessary to compute the interpolation functions and r^2 coefficients may appear to be a cumbersome task, the method is easily carried out with

the use of a computer. Appendix F completely outlines the Maple commands that are used to carry the computations for this simple example.

Table 5.1. r^2 Coefficient Values for Various Degree Curve Fitting Function.

Polynomial Order	r^2 Value
2	0.9923
3	0.9998
4	0.9999
12	0.9999

CHAPTER VI

Phase One Optimization

Introduction

As already stated, the objective of phase one is to determine the optimal pole configuration for a specified stroke position. Recall that the pole configuration of the solenoid is characterized by the notch angle, A , and notch depth, D , as depicted in Figure 6.1.

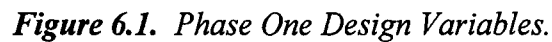
The first section of this chapter outlines the development of the phase one force function having the general form,

$$force = f(S, D, A) \quad (6.1)$$

where S is stroke position, D is notch depth and A is notch angle. The section will discuss several techniques for assessing the accuracy of the curve fitting function produced by the method of least squares. In the final section, a method will be clearly defined for using the curve fitting force function to determine optimal pole configurations.

Developing a Force Function

The methods used to develop an algebraic expression of force is similar to that presented in the example least squares problem described of Chapter V. The main difference is the number of independent variables within the curve fit function.



F = [...data...], **S** = [...data...], **A** = [...data...], **D** = [...data...]

Chapter IV also describes the manner in which design parameters were varied for the purpose of creating an appropriately descriptive data set. This data set includes results for a range of S , D , and A values that allow for a sufficiently large range of

optimal pole configurations. Table 4.4 summarizes the range of values for the notch angle, A , and notch depth, D , that were analyzed at five different stroke positions. In all, a total of 65 finite element models are created and analyzed in order to produce the four data arrays.

After analysis is complete, the effect of pole configuration may quickly be characterized upon examining the various plots of the finite element results. For example, Figure 6.2 shows a plot *Force vs. Notch Angle* for various stroke positions. By examining this plot, some conclusions may be drawn regarding optimal notch angles for different stroke positions (i.e. smaller notch angle for larger stroke distance is optimal). Similarly, a plot of *Force vs. Notch Depth* (Figure 6.3) may yield useful information about optimal notch depths. Lastly, such plots may be used to verify that the finite element results hold true with expected results. For example, Figure 6.4 shows the typical nonlinear relationship between force and stroke position. (Note: In Figures 6.2-6.4, lines are drawn between data points for visual purpose only. These lines do not indicate a linear interpolation of data between data points).

While these plots are useful in describing the effect of pole configuration, the stated objective is to represent the data set with a single algebraic function having the form of (6.1). This is achieved by using the method of least squares to curve fit a function to the data set. As in the example problem of Chapter V, the first step of the least squares Maple algorithm is to input the data set in the form of arrays. Before inputting, the data is normalized by dividing each entry by a common normal value that corresponds to the respective array. By lowering the range of values in each array, computer round-off error

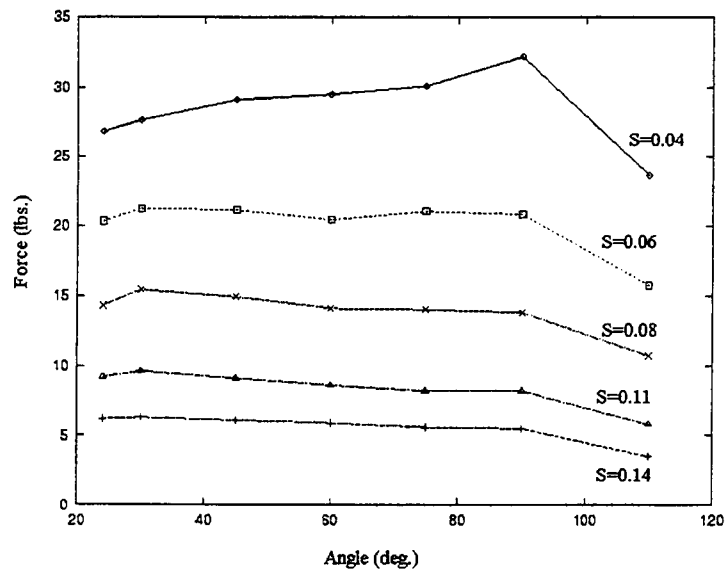


Figure 6.2. Plot of Finite Element Results Showing Force vs. Notch Angle.

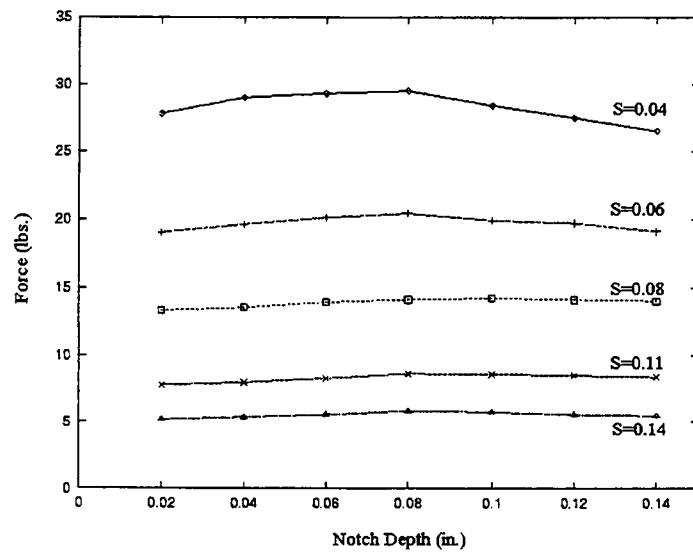


Figure 6.3. Plot of Finite Element Results Showing Force vs. Notch Depth.

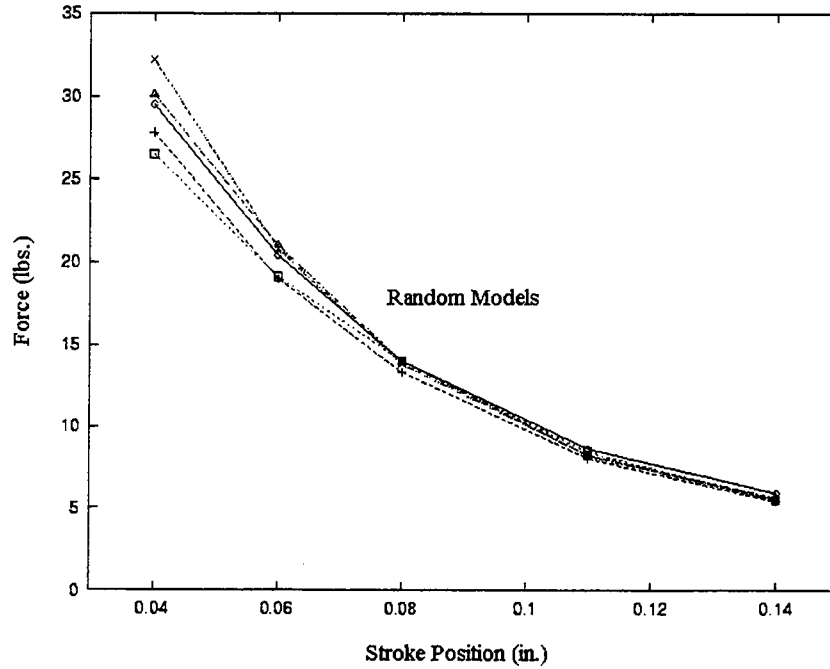


Figure 6.4. Plot of Finite Element Results Showing Force vs. Stroke Position.

is reduced [12]. (Note: Normalizing values are noted within the complete Maple worksheet given in Appendix G).

After the data set is input, the next step is to establish the form of the curve fitting. The form of the curve fitting function is obtained by experimentation within the Maple worksheet until a sufficiently accurate interpolation is achieved. As will be discussed, there are a number of techniques that may be used to assess the accuracy of curve fitting functions. The form of the curve fitting function used in phase one is given as,

$$\begin{aligned} \text{force} = & c_1 + c_2 S + c_3 S^2 + c_4 S^3 + c_5 S^4 + c_6 D + c_7 D^2 + c_8 D^3 \\ & + c_9 D^4 + c_{10} D \cdot S + c_{11} A + c_{12} A^2 + c_{13} A^3 + c_{14} A^4 + c_{15} A \cdot S \end{aligned} \quad (6.2)$$

where $c_{1 \rightarrow 15}$ represent the coefficients to be determined by the method of least squares.

Similar to the example problem in Chapter V, the problem consists of determining these constants from satisfying the condition that,

$$Q = \sum_{i=1}^{65} \eta_i^2 \quad (6.3)$$

is minimized. This sum is carried out over each of the 65 data points that were obtained from phase one finite element analysis. By differentiating (6.3) with respect to each of the 15 constants, the result is 15 equations that are set equal to zero. As an example, the derivative of Q with respect to c_8 is given in (6.4).

$$\begin{aligned} \frac{\partial Q}{\partial c_8} = -2 \sum_{i=1}^{65} D_i^3 (force_i - c_1 - c_2 S_i - c_3 S_i^2 - c_4 S_i^3 - c_5 S_i^4 - c_6 D_i - c_7 D_i^2 - c_8 D_i^3 \\ - c_9 D_i^4 - c_{10} D_i S_i - c_{11} A_i - c_{12} A_i^2 - c_{13} A_i^3 - c_{14} A_i^4 - c_{15} A_i S_i) = 0 \end{aligned} \quad (6.4)$$

The 15 equations are then solved simultaneously for each of the 15 unknown coefficients to produce the following phase one force function.

$$\begin{aligned} force = 2.56 - 4.54S + 2.49S^2 - 0.64S^3 + 0.07S^4 - 0.05D + 0.30D^2 \\ - 0.32D^3 + 0.09D^4 + 0.06DS + 4.58A - 7.50A^2 + 5.15A^3 - 1.25A^4 \end{aligned} \quad (6.5)$$

where the values of the constants are rounded to the nearest 0.01. The entire least squares method is carried out within a single Maple worksheet given in Appendix G.

Assessing the Quality of the Curve Fitting Function

Information regarding the accuracy of the curve fitting function is obtained from the r^2 value (coefficient of determination). As described in Chapter V, the r^2 value is a quantitative assessment of how well the curve fits the data set. The r^2 coefficient for the phase one interpolating function was found to be 0.999 (see Appendix G). Since this value is close to one, it is evident that the function sufficiently approximates the data set.

The second method that is used to test the accuracy of the interpolation function is random testing. Random testing refers to the finite element analysis of a random pole configuration followed by a force calculation for the same configuration using the curve fitting function. By comparing the two force values, one is able to gain information regarding the quality of the curve fitting function. This is carried out a number of times to ensure the quality of the curve fitting functions.

A major benefit of random tests is that they are capable of identifying instabilities within the curve fitting function. As described in Chapter V, these instabilities come in the form of oscillations between data points. For this reason, random tests were performed for design variable values between those used in the original finite element analysis. The results of the random tests that were performed are summarized in Table 6.1. Notice that the percent difference between the curve fit and finite element force value is sufficiently accurate. Also, since these random configurations are between original finite element configurations, it is safe to assume that oscillations are not present and the thus curve fitting function is stable.

Table 6.1. Results of Phase One Random Tests.

S (in.)	D (in.)	A (deg.)	FEA Force (lb.)	Curve Fit Force (lb.)	% Difference
0.05	0.05	35	36.554	34.632	5.55
0.10	0.07	55	14.286	14.399	0.78
0.13	0.11	80	10.251	9.806	4.54

Phase One Optimization

The mathematics involved in determining the optimal pole configuration is best understood by examining a plot of the curve fitting function. Since the pole configuration is to be optimized at a specified stroke position, the optimizing function is obtained from the equation,

$$force = f(D, A, S = \text{constant}) \quad (6.6)$$

where a normalized value for stroke position is substituted into (6.5). For example, substituting the normalized value corresponding to a stroke position of 0.04 inches into (6.5) gives the following function.

$$force = -0.53 + 4.52A - 7.50A^2 + 5.14A^3 - 1.25A^4 + 0.04D + 0.30D^2 - 0.32D^3 + 0.09D^4 \quad (6.7)$$

Figure 6.5 shows a two dimensional plot of *Force vs. Notch Angle and Notch Depth* that is obtained from (6.7). As pointed out in the figure, it is obvious that two local maxima exist over the particular range that has been plotted. The goal is to

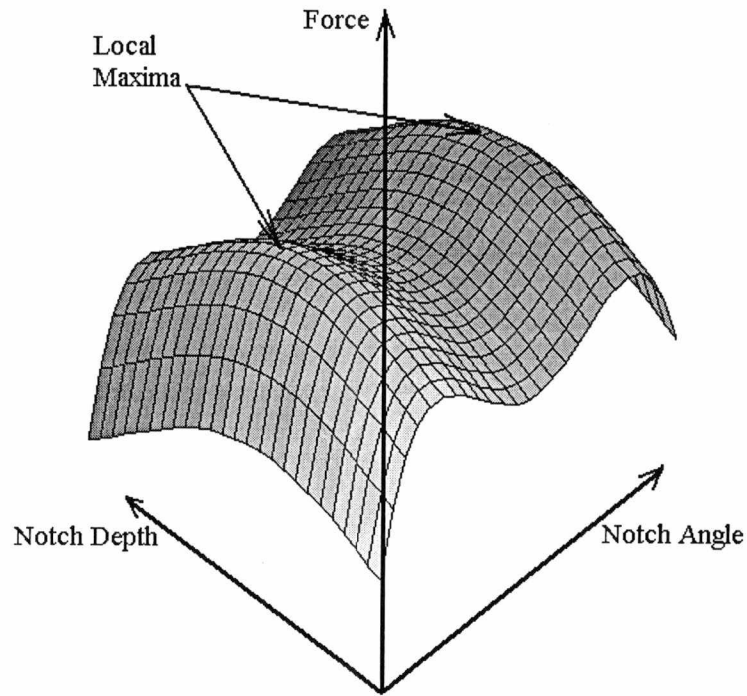


Figure 6.5. Surface Plot of Force vs. Notch Angle and Notch Depth.

mathematically determine the *single* maximum force value that exists over this surface, and its corresponding A and D values. This set of values corresponds to the optimal pole configuration for a stroke position of 0.04 inches.

The local maxima, local minima and points of inflection for (6.7) exist where the first derivatives with respect to A and D are equal to zero. This condition is given by the following equations.

$$\frac{\partial \text{Force}}{\partial A} = 4.52 - 15.0A + 15.44A^2 - 5.00A^3 = 0 \quad (6.8)$$

$$\frac{\partial \text{Force}}{\partial D} = 0.04 + 0.61D - 0.97D^2 + 0.34D^3 = 0 \quad (6.9)$$

Solving these equations for A and D produces the following nine pairs of roots that correspond to the maxima, minima and inflection points for the surface plot.

$$\text{Root_1} = [A = 36.8^\circ, D = 0.0004 \text{ in.}]$$

$$\text{Root_2} = [A = 61.2^\circ, D = 0.0005 \text{ in.}]$$

$$\text{Root_3} = [A = 86.9^\circ, D = 0.0004 \text{ in.}]$$

$$\text{Root_4} = [A = 36.8^\circ, D = 0.073 \text{ in.}]$$

$$\text{Root_5} = [A = 61.4^\circ, D = 0.073 \text{ in.}]$$

$$\text{Root_6} = [A = 86.9^\circ, D = 0.073 \text{ in.}]$$

$$\text{Root_7} = [A = 36.8^\circ, D = 0.154 \text{ in.}]$$

$$\text{Root_8} = [A = 61.2^\circ, D = 0.154 \text{ in.}]$$

$$\text{Root_9} = [A = 86.9^\circ, D = 0.154 \text{ in.}]$$

By computing the second derivatives of (6.7) with respect to A and D , it can be determined whether these roots correspond to local maxima, local minima, or points of inflection. The second derivatives of (6.7) are given as

$$\frac{\partial^2 \text{force}}{\partial A^2} = -15.00 + 30.88A - 15.01A^2 \quad (6.10)$$

$$\frac{\partial^2 \text{force}}{\partial D^2} = 0.61 - 1.94D + 1.02D^2 \quad (6.11)$$

A negative second derivative indicates a ‘concave down’ shape and hence a local maxima. By substituting the A and D values from the nine roots into (6.10) and (6.11), respectively, the following nine pairs of values are obtained.

$$\text{Pair_1} = [-1.72, 0.59]$$

$$\text{Pair_2} = [0.87, 0.59]$$

$$Pair_3 = [-1.77, 0.59]$$

$$Pair_4 = [-1.72, -0.31]$$

$$Pair_5 = [0.87, -0.31]$$

$$Pair_6 = [-1.77, -0.31]$$

$$Pair_7 = [-1.71, 0.66]$$

$$Pair_8 = [0.87, 0.66]$$

$$Pair_9 = [-1.77, 0.66]$$

Two negative values within a pair indicate that the pair corresponds to a local maximum root for both A and D . For example, the only two pairs of this set that contain two negative numbers are $Pair_4$ and $Pair_6$, corresponding to $Root_4$ and $Root_6$ of the previous list of roots. The A and D values of these roots produce force values corresponding to the two local maximums shown in Figure 6.5. The exact force values given below are obtained by substituting the A and D values of each root into (6.7).

$$Force_4 = 29.44 \text{ lb.}$$

$$Force_6 = 29.51 \text{ lb.}$$

The final step is to examine the force values produced by local maxima roots to determine which is greater. For this example, it is obvious that the $Force_6$ value is greater and thus $Root_6$ ($A = 86.9^\circ$, $D = 0.073 \text{ in}$) corresponds to the optimal pole configuration values for a stroke position of 0.04 inches.

As a second example, the optimal pole configuration for a stroke position of 0.12 inches is determined. Again, the function to be optimized is obtained by substituting the corresponding stroke position value into (6.7) to produce the following equation.

$$\begin{aligned} \text{force} = & -0.97 + 4.54A - 7.50A^2 + 5.15A^3 - 1.25A^4 \\ & + 0.01D + 0.30D^2 - 0.32D^3 + 0.09D^4 \end{aligned} \quad (6.12)$$

Carrying out the same process as described in the previous example produces the two local maxima roots given below.

$$\text{Root}_2 = [A = 35.76^\circ, D = 0.085 \text{ in.}]$$

$$\text{Root}_8 = [A = 85.7^\circ, D = 0.085 \text{ in.}]$$

Substituting these roots into (6.12) produces the following force values from which it is evident that *Root_2* corresponds to the optimal pole configuration for a stroke position of 0.14 inches.

$$\text{Force}_2 = 11.79 \text{ lbs.}$$

$$\text{Force}_8 = 11.29 \text{ lbs.}$$

Further Analysis

As mentioned in Chapter II, it has been documented that a notch angle of 30° is optimal for stroke positions of 0.06 inches or larger. It is also known that a flat-faced configuration (notch angle of 90°) produces optimal results for most stroke positions less than 0.06 inches [11]. Plots may be used to determine whether the curve fitting function optimizations produces results that correspond to these observations.

For example, the curve fitting force function is plotted at a stroke position of 0.04 inches and a notch depth of 0.08 inches in Figure 6.6. As expected, the optimal force occurs near a notch angle of 90° . When a stroke position of 0.14 inches is supplied to the force function and plotted, the optimal angle is observed to be 30° (Figure 6.7). The fact

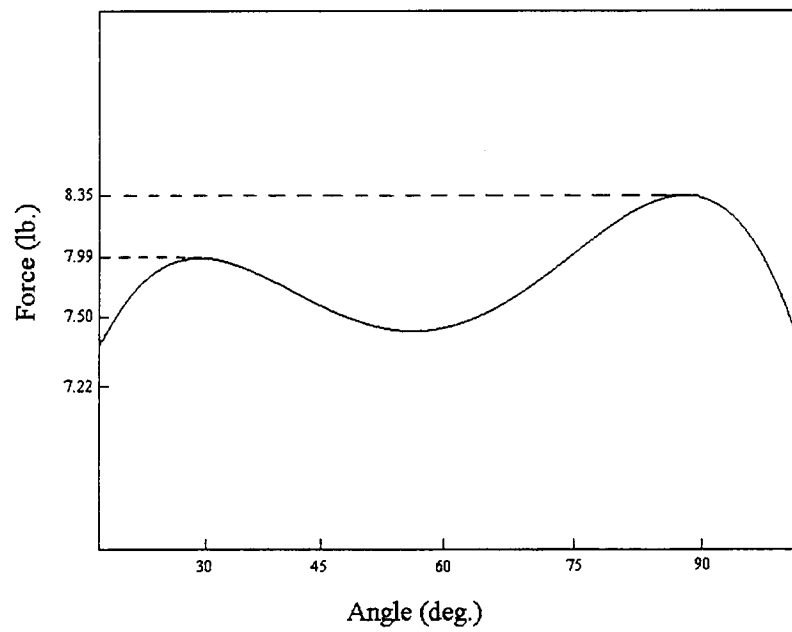


Figure 6.6. Force vs. Notch Angle for Stroke Position of 0.04 inches.

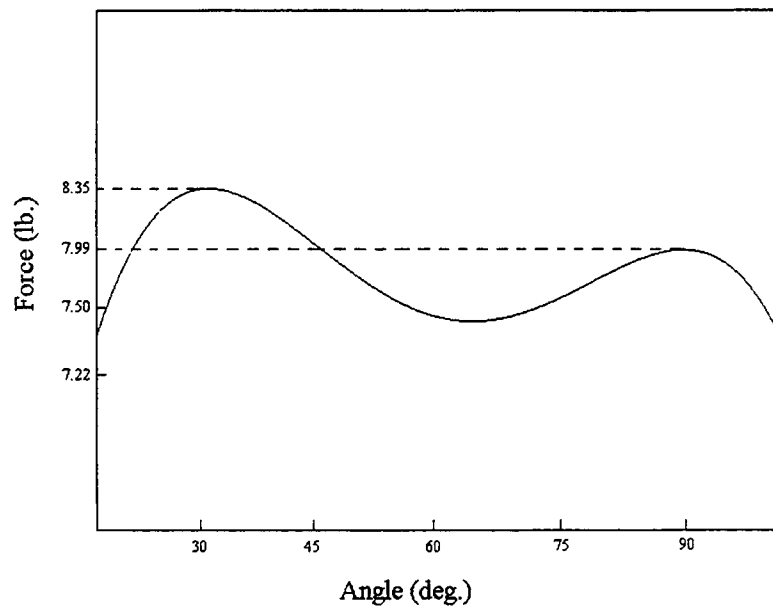


Figure 6.7. Force vs. Notch Angle for Stroke Position of 0.14 inches.

that these plots agree with expected results are further evidence of the quality of the interpolating function.

Applying Information

For the cost-dependent study of Chapter VIII, a flat-faced pole configuration ($A = 90^\circ$, $D = 0.08$ in.) is used for all finite element analysis. This is because it is impractical to produce a finite element database for each optimal pole configuration. Once a cost-optimal design is determined for a particular force requirement and stroke position, the optimal pole configuration will be applied in order to supply an additional force. Since this is a no-cost improvement in the design, no further cost optimization is required. The next two chapters discuss the solenoid cost functions and phase two optimization that is used to identify the cost-optimal solenoid.

CHAPTER VII

Presentation of Cost Equations

The previous chapter outlined the development of algebraic expressions of solenoid force output as a function of the set of design variables. In order to optimize the force output of the solenoid with respect to cost, an algebraic expression of cost must also be developed. This chapter presents the equations that express solenoid cost as a function of its geometry. The task is to express solenoid cost as an algebraic function of same design variables considered in the phase two force analysis,

$$Cost = f(L, C_R, P_R, W) \quad (7.1)$$

where L is the length of the solenoid; C_R is the outside case radius, P_R is the plunger radius, and W is the case wall thickness (see Figure 3.1).

In order to develop (7.1), the costs of all components of the solenoid are written as functions of geometry. For the purpose of developing a cost function, the solenoid is analyzed as four separate components. These components are; 1) washers, 2) steel casing, 3) steel core and plunger, and 4) copper coil. An expression of the form (7.1) is then achieved by combining the cost expressions for each of these four components. The following sections present previously developed cost functions for these four components. These equations were developed by Dr. Frank Speckhart and are used with permission for the purpose of this optimization study [16].

Washer Costs

Two steel washers are used at the top and bottom of the small push-pull solenoid assembly. Both washers are identical and act to hold the solenoid together at the top and bottom. The diameter of the washers is closely approximated as the overall solenoid diameter. While there exists magnetic flux through the washers, it can be shown from finite element analysis that the overall effect of washer thickness is small. For this reason, washer thickness is assumed as a constant equal to 0.083 inches. The following equation is then used to evaluate the total cost in dollars of both washers.

$$WasherCost = (0.02 + 0.30 \cdot (WasherVolume)) \quad (7.2)$$

The volume to be substituted into this expression does not take into account the hole and thus the volume of a washer is then given by the expression,

$$WasherVolume = \pi \cdot C_R^2 \cdot (0.083) \quad (7.3)$$

where C_R is the outer radius of the solenoid case. By substituting (7.3) into (7.2) and multiplying by two, the total cost of both washers is found to be,

$$TotalWasherCost = 0.02 + 0.30 \cdot (\pi \cdot C_R^2 \cdot (0.083)) \quad (7.4)$$

Steel Casing Cost

The case is the steel cylinder that surrounds the solenoid. The following expression is used to evaluate the cost of a single solenoid case.

$$CaseCost = 0.36 \cdot (CaseVolume) + 0.08 \cdot C_R + 0.26 \quad (7.5)$$

The volume of the case is given by,

$$CaseVolume = (L + 0.15) \cdot \left(\frac{\pi}{4} \cdot [OD^2 - ID^2] \right) \quad (7.6)$$

where L is also a design variable that represents the length of the case in inches. Notice that 0.15 inches is added to the length to account for wasted material. The OD and ID in the equation refer to the outside and inside diameter of the case. Realizing that

$OD = 2 \cdot C_R$ and that $ID = 2 \cdot (C_R - W)$ and substituting (7.6) into (7.5) an expression for the cost of a single steel case is found to be,

$$CaseCost = 2.262 \cdot L \cdot C_R - 1.131 \cdot L \cdot W + 0.045 \cdot C_R \cdot W - 0.022 \cdot W^2 + 0.08 \cdot C_R + 0.26 \quad (7.7)$$

Core and Plunger Cost

The core and plunger are found at the center of the coil of the solenoid. Since both parts must have equal diameters, they require only a single cost expression. The cost of a solenoid core and plunger is given as,

$$CorePlungerCost = 0.25 \cdot CorePlungerVolume + 0.10 \cdot P_R + 0.05 \quad (7.8)$$

where P_R is the radius of the plunger. The volume of the core/plunger is given by the expression,

$$CorePlungerVolume = (L + 0.15) \cdot \left(\frac{\pi}{4} \cdot OD^2 \right) \quad (7.9)$$

where $(L + 0.15)$ is used again as the effective length of the solenoid. Realizing that $OD = 2P_R$ and substituting (7.9) into (7.8) gives the following expression for the total cost of the solenoid plunger and core:

$$CorePlungerCost = 0.785 \cdot L \cdot P_R \quad (7.10)$$

Coil Cost

The final solenoid component to be analyzed with respect to cost is the copper coil. The following expression describes the cost of copper within the solenoid.

$$CopperCost = 0.78 + 0.90(CopperVolume) \quad (7.11)$$

The volume of copper in the solenoid is given as:

$$CopperVolume = (L - 0.166) \cdot \left(\frac{\pi}{4} \cdot [OD^2 - ID^2] \right) \quad (7.12)$$

where OD and ID refer to the outside and inside diameters of the copper coil. Again 0.02 inches is added to the length to account for waist material. Realizing that

$OD = 2 \cdot (C_R - W)$ and $ID \approx 2 \cdot P_R$, the cost of the copper coil can be rewritten as

$$TotalCopperCost = 0.78 + 2.83 \cdot LC_R^2 - 5.65 \cdot LWC_R + 2.83 \cdot LW^2 - 2.83 \cdot LP_R^2 \quad (7.13)$$

Total Solenoid Cost

The total solenoid cost is obtained by adding the costs of each component which is written generally as,

$$\begin{aligned} \text{SolenoidCost} = & \text{TotalWasherCost} + \text{CaseCost} \\ & + \text{Core / PlungerCost} + \text{CopperCoilCost} \end{aligned} \quad (7.14)$$

Adding these terms and simplifying gives the following expression of total solenoid cost as a function of each cost-effecting design variable,

$$\begin{aligned} \text{SolenoidCost} = & 1.11 + 0.08 \cdot C_R^2 - 3.39 \cdot LWC_R + 0.45 \cdot WC_R + 1.70 \cdot LW^2 \\ & - 0.23 \cdot W^2 + 0.02 \cdot C_R - 2.04 \cdot LP_R^2 + 0.16 \cdot P_R^2 + 0.10 \cdot P_R + 2.83 \cdot LC_R^2 \end{aligned} \quad (7.15)$$

The calculations used to compute these component costs are given as a Maple worksheet in Appendix H. The total solenoid cost equation (7.15) is used extensively in the following chapters to predict cost-optimal solenoid configurations.

CHAPTER VIII

Phase Two Optimization

Introduction

While the pole configuration optimization of phase one will undoubtedly add to the overall solenoid performance, the major design parameters of the solenoid are now considered. In this chapter, the phase two design variables are chosen to produce a complete cost-optimal solenoid design. Figure 8.1 once again shows the design variables to be optimized in phase two. These include case radius, C_R , length, L , plunger radius, P_R , and case wall thickness, W .

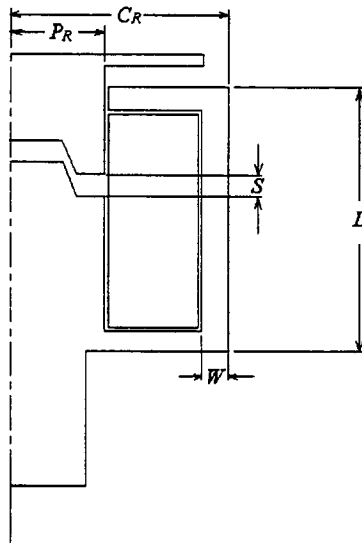


Figure 8.1. Phase Two Design Variables.

The objective of phase two is to provide the cost-optimal solenoid design for a specified minimum force requirement at a particular stroke position. Once this optimal design is determined, the configuration is combined with the corresponding optimal phase one pole configuration to produce a complete design. This chapter outlines the methods used to develop an algebraic expression of force with respect to the phase two design variables. Secondly, this chapter discusses how the previously developed cost equation is used in conjunction with the phase two force function to identify cost-optimal values. In the next chapter, a number of examples will be considered that outline these steps for real design problems.

Developing a Force Function

As in phase one, the first step in the optimization process is to develop an algebraic expression for force having the general form,

$$force = f(C_R, L, P_R, W, S). \quad (8.1)$$

Again, this accomplished by curve fitting a function to the database of finite element results that was described in Chapter IV. In phase two, force is determined as a function of five variables and thus requires the development of the six data arrays depicted below.

$$\mathbf{F} = [\dots data \dots], \quad \mathbf{C_R} = [\dots data \dots], \quad \mathbf{L} = [\dots data \dots]$$

$$\mathbf{W} = [\dots data \dots], \quad \mathbf{P_R} = [\dots data \dots], \quad \mathbf{S} = [\dots data \dots]$$

As in phase one, these data arrays consist of finite element results. The finite element analysis process for phase two is described in detail in Chapter IV. Analysis is performed over predetermined ranges for each variable, and at four stroke positions

resulting in 104 data points. The geometric range of analysis for each of the phase two design variables is summarized in Table 4.7.

As with phase one, it is useful to examine plots of the finite element results to gain a perspective on the effects of each design variable on force performance. Plots of the finite element results are given in Figures 8.2-8.5. Each plot shows the effect of a particular design variable on force performance for various stroke positions (Note: Lines drawn between data points are for visual purposes only. These lines are not intended to represent a linear interpolation between data points).

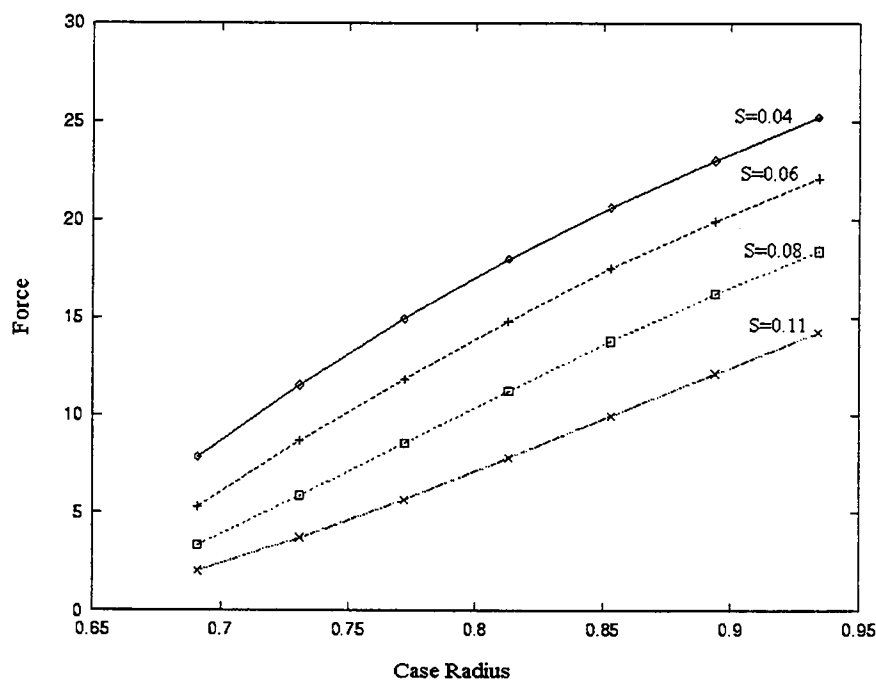


Figure 8.2. Plot of Finite Element Results Showing Force vs. Case Radius.

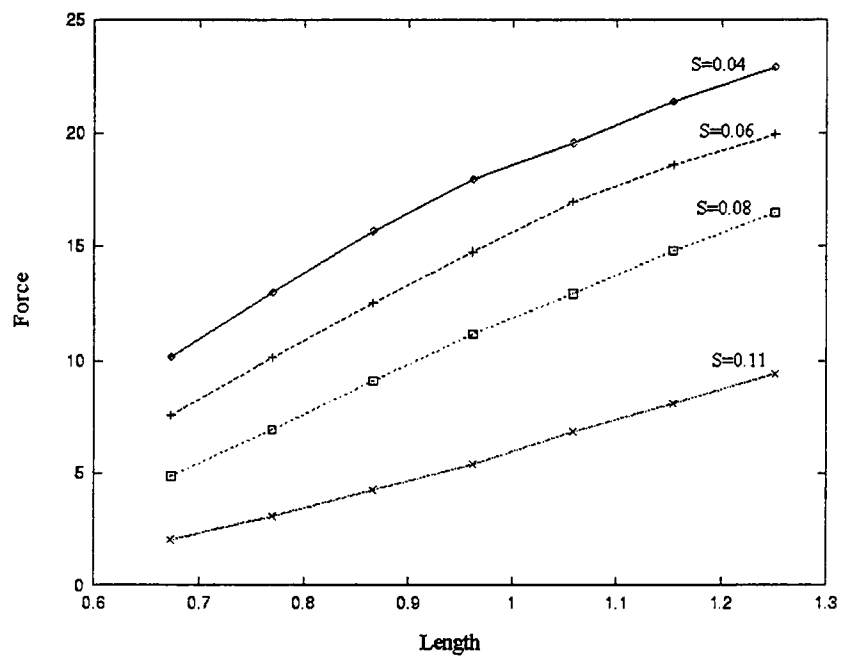


Figure 8.3. Plot of Finite Element Results Showing Force vs. Length.

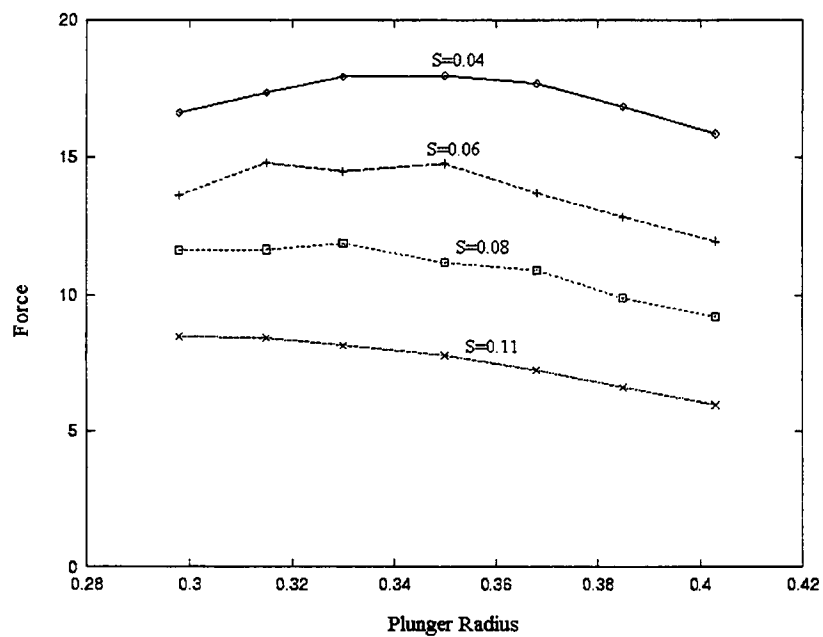


Figure 8.4. Plot of Finite Element Results Showing Force vs. Plunger Radius.

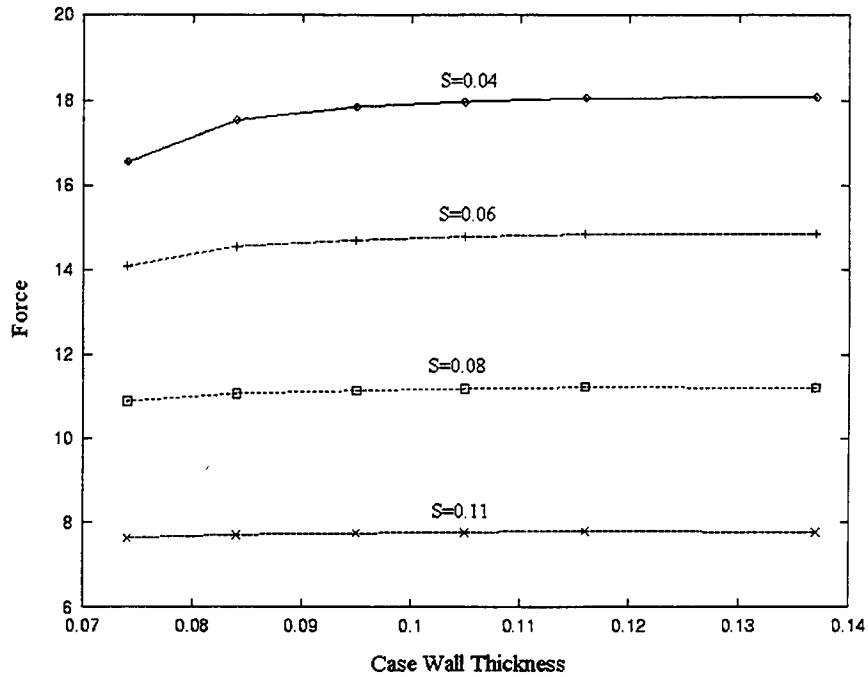


Figure 8.5. *Plot of Finite Element Results Showing Force vs. Case Wall Thickness.*

These plots are useful in determining the form of the curve fitting function that should be used. For example, Figure 8.2 shows that the effect of solenoid length on force performance is almost perfectly linear. Similarly, Figure 8.3 shows a generally linear relationship between force and case radius. However, the curves depicted in Figure 8.4, and Figure 8.5 indicate that the interpolation function should be at least quadratic in P_R and W .

Perhaps an appropriate interpolating force function would be a linear combination of terms whose order is based on the previous observations. This expression would have the general form,

$$force = c_1 + c_2L + c_3C_R + c_4P_R + c_5P_R^2 + c_6W + c_7W^2 + c_8S + c_9S^2 \quad (8.2)$$

where L and C_R are first degree, while P_R , W , and S all have quadratic terms.

A curve fitting function of this form is easily obtained by using the method of least squares in conjunction with the phase two data arrays. The explicit expression for force having this form is calculated to be (see Appendix I),

$$force = -6.2 + 1.36L + 4.83C_R + 17.75P_R - 27.56P_R^2 - 0.27W + 3.71W^2 - 1.13S + 0.12S^2 \quad (8.3)$$

Since the method of least squares for curve fitting has already been described in Chapters V and VI, it is not discussed here. Once (8.3) is established, the coefficient of determination (r^2 value) can be used to assess how well the function approximates the data set. The r^2 value for the function of this form is calculated to be 0.978, which is considered to be an accurate fit [4] (see Appendix I).

After experimenting with several forms, another function was developed having the form,

$$force = c_1 + c_2L + c_3L^2 + c_4C_R + c_5C_R^2 + c_6P_R + c_7P_R^2 + c_8W + c_9W^2 + c_{10}S + c_{11}S^2 \quad (8.4)$$

where all variables are represented by a second order term. The method of least squares produces the following explicit expression of force (see Appendix J).

$$force = -9.72 - 4.38L - 1.48L^2 + 7.69C_R - 1.73C_R^2 + 21.46P_R - 32.86P_R^2 + 2.72W - 10.51W^2 - 1.12S + 0.11S^2 \quad (8.5)$$

The r^2 value for this curve fitting force function is calculated to be 0.982, which is slightly better than the previous function (see Appendix J). For this reason, (8.5) is chosen to be the force function for phase two.

As with the phase one force function, random tests are performed to further assess the accuracy of the phase two curve fitting function given in (8.5). Again, random tests involve performing finite element analysis on random configurations and comparing the resulting force values with those calculated from the curve fitting function. Plots of the random test results are given in Figures 8.6-8.8 and Table 8.1 summarizes the actual data that was obtained. (Note: Lines drawn between data points are for visual purposes only. These lines are not intended to represent a linear interpolation between data points).

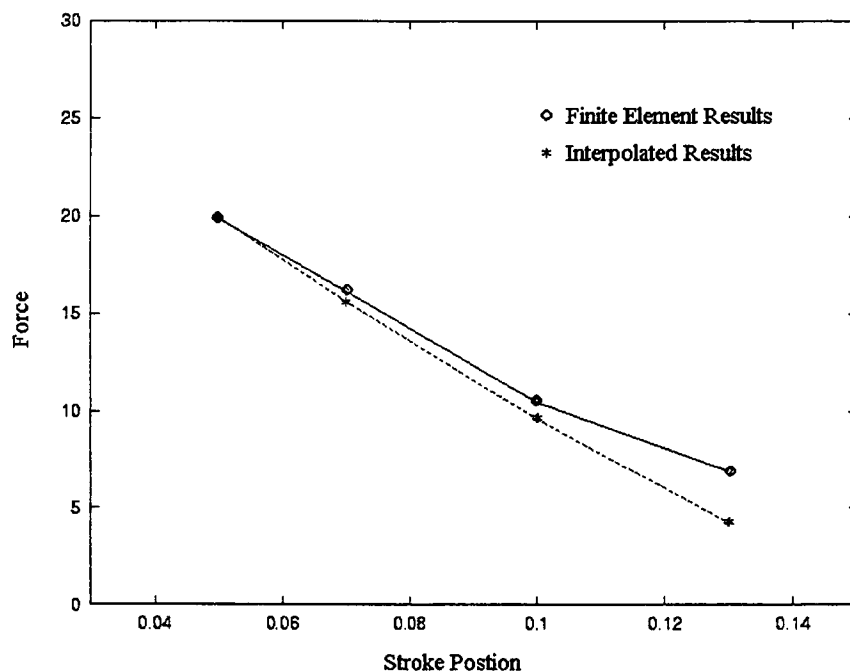


Figure 8.6. Random Test #1 Results Showing Force vs. Stroke Position.

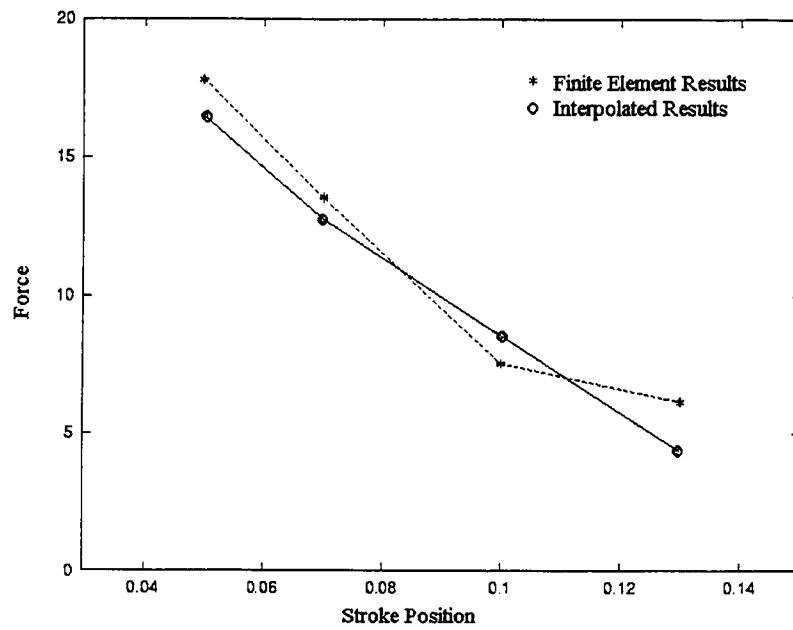


Figure 8.7. Random Test #2 Results Showing Force vs. Stroke Position.

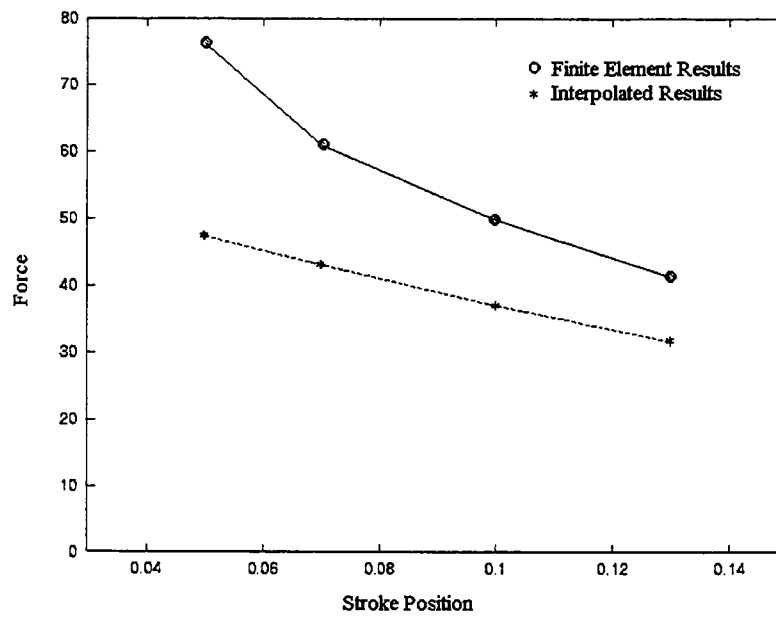


Figure 8.8. Random Test #3 Results Showing Force vs. Stroke Position.

Table 8.1. Results of Phase Two Random Tests.

C_R	P_R	W	L	S	FE Force (lb.)	Interp. Force (lb)	% Error
0.900	0.390	0.125	0.724	0.05	19.845	19.914	0.35
0.900	0.390	0.125	0.724	0.07	16.133	15.604	3.28
0.900	0.390	0.125	0.724	0.10	10.481	9.629	8.13
0.900	0.390	0.125	0.724	0.13	6.930	4.242	38.79
0.710	0.300	0.079	1.200	0.05	16.492	17.822	8.06
0.710	0.300	0.079	1.200	0.07	12.739	13.512	6.07
0.710	0.300	0.079	1.200	0.10	8.550	7.537	11.85
0.710	0.300	0.079	1.200	0.13	6.256	2.150	65.63
1.100	0.420	0.140	1.320	0.05	76.267	47.337	37.93
1.100	0.420	0.140	1.320	0.07	61.035	43.027	29.50
1.100	0.420	0.140	1.320	0.10	49.819	37.052	25.63
1.100	0.420	0.140	1.320	0.13	41.240	31.665	23.22

Upon examination of Table 8.1 and the corresponding plots, it is obvious that some of the random configurations are more accurate than others. Notice that the curve fit force values for configurations that fall within the original finite element analysis range are very near those calculated from the finite element method. Conversely, the percent difference is much higher for configurations that are not within the analysis range. For example, the curve fit data for a stroke position of 0.13 inches is different from the force obtained from the finite element method. This inaccuracy is due to the fact that these particular data points ($S = 0.13$) were extrapolated beyond the stroke range that was used to produce the database of finite element results (Recall that each configuration in phase two is analyzed at stroke positions of 0.04, 0.06, 0.08, and 0.11 inches).

Figure 8.8 shows a plot of a random test configuration in which each design variable exceeds the finite element analysis range. As expected, the curve fit function produces results that are far different from those produce from finite element analysis [10]. From this evidence, it can be safely stated that the interpolation function is only accurate over the original test range. This will become an especially important fact when exploring techniques for determining the cost-optimal configuration.

Introduction to Phase Two Optimization

Now that an explicit expression of force as a function of the phase two design variables has been obtained, it must be used to identify the cost-optimal design. Unlike phase one, however, this does not simply involve finding the maximum of the force function over a given range. Instead, the problem consists of minimizing the cost function while meeting a specified force at a given stroke position.

As developed in Chapter VII, solenoid cost is expressed with the following equation which is a function of the same phase two design variables.

$$\begin{aligned} \text{SolenoidCost} = & 1.11 + 0.08 \cdot C_R^2 - 3.39 \cdot LWC_R + 0.45 \cdot WC_R + 1.70 \cdot LW^2 \\ & - 0.23 \cdot W^2 + 0.02 \cdot C_R - 2.04 \cdot LP_R^2 + 0.16 \cdot P_R^2 + 0.10 \cdot P_R + 2.83 \cdot LC_R^2 \end{aligned} \quad (8.6)$$

As stated, the goal is to minimize (8.6) while satisfying an expression of the form,

$$\begin{aligned} Mforce = & -9.72 - 4.38L - 1.48L^2 + 7.69C_R - 1.73C_R^2 + 21.46P_R \\ & - 32.86P_R^2 + 2.72W - 10.51W^2 - 1.12S + 0.11S^2 \end{aligned} \quad (8.7)$$

where S is replaced in (8.5) by the stroke position being considered and $Mforce$ is the specified minimum force value to be exerted by the solenoid at the position, S .

The following section proposes the mathematical solution of this problem through the use of the *penalty method*.

Penalty Method

The penalty method is used here to mathematically minimize the cost function while maintaining the specified minimum force requirement in the form of a constraint equation. In doing so, the penalty equation is introduced which has the form,

$$Penalty = SolCost - \lambda (Mforce - force) \quad (8.8)$$

where *SolCost* is the solenoid cost function, *Mforce* is the minimum force requirement, *force* is the force function, and λ is a constant to be determined.

Notice that the constraint term, $\lambda (Mforce - force)$, in (8.8) is equal to zero for configurations that exactly produce the minimum force requirement. Consequently, a value of zero is added to the cost term for configurations produce the minimum force requirement. This means that all minimum values of the penalty function will occur only where the minimum force requirement is produced. As a result, traditional calculus can be used to minimize (8.8) and thus determine those design configurations that produce the minimum force for the lowest cost.

For example, suppose that one is to use the penalty method to determine the optimal configuration for a minimum force of 13.0 lb. at a stroke position of 0.10 inches. Substituting the actual functions corresponding to *SolCost* and *force* into (8.8) produces the following equation.

$$penalty = 1.11 + .08C_R^2 - 3.39C_RWL + 0.45C_RW + 1.70W^2L - 0.23W^2 + 0.02C_R \quad (8.9)$$

$$\begin{aligned}
penalty = & 1.11 + .08C_R^2 - 3.39C_RWL + 0.45C_RW + 1.70W^2L - 0.23W^2 + 0.02C_R \quad (8.9) \\
& - 2.04P_R^2L + 0.16P_R^2 + 0.10P_R + 2.83LC_R^2 - \lambda(Mforce - 9.72 - 1.11S + 0.11S^2 \\
& - 7.69C_R - 1.73C_R^2 + 2.73W - 10.51W^2 + 4.38L - 1.4L^2 + 21.46P_R - 32.86P_R^2)
\end{aligned}$$

The normalized minimum force and stroke position values are substituted into (8.9) to give,

$$\begin{aligned}
penalty = & 1.11 + .08C_R^2 - 3.39C_RWL + 0.45C_RW + 1.70W^2L - 0.23W^2 + 0.02C_R \quad (8.10) \\
& - 2.04P_R^2L + 0.16P_R^2 + 0.10P_R + 2.83LC_R^2 - \lambda(10.82 - 7.69C_R - 1.73C_R^2 \\
& + 2.73W - 10.51W^2 + 4.38L - 1.4L^2 + 21.46P_R - 32.86P_R^2)
\end{aligned}$$

Taking partial derivatives of (8.10) with respect to C_R , P_R , L , W , and λ produces five equations which are set then set equal to zero to give the following.

$$\frac{\partial penalty}{\partial C_R} = 0.02 + 0.15C_R - 3.39WL + 0.45W + 5.65LC_R + 7.69\lambda - 3.46\lambda C_R = 0 \quad (8.11)$$

$$\frac{\partial penalty}{\partial P_R} = 0.10 + 0.31P_R - 4.08P_RL + 21.46\lambda - 65.72\lambda P_R = 0 \quad (8.12)$$

$$\frac{\partial penalty}{\partial L} = -3.39WC_R + 1.70W^2 - 2.04P_R^2 + 2.83C_R^2 + 4.38\lambda - 2.95\lambda L = 0 \quad (8.13)$$

$$\frac{\partial penalty}{\partial W} = 0.45C_R - 3.39LC_R + 3.39WL - 0.45W + 2.72\lambda - 21.03\lambda W = 0 \quad (8.14)$$

$$\begin{aligned}
\frac{\partial penalty}{\partial \lambda} = & -22.72 + 7.69C_R - 1.73C_R^2 + 2.72W - 10.51W^2 \quad (8.15) \\
& + 4.38L - 1.48L^2 + 7.69\lambda - 3.46\lambda C_R = 0
\end{aligned}$$

These five equations are then solved simultaneously for C_R , P_R , L , W , and λ .

This produces a large number of roots, many of which are imaginary. The Maple

worksheet given in Appendix J includes a function for determining real roots only, which is calculated to be,

$$\text{Real Root} = [CR=0.78, L=0.65, PR=0.34, W=0.20, \lambda = -0.42]$$

Notice that the solenoid length, L , and outside case thickness, W , are values that lie outside the original finite element analysis range. As already stated, any extrapolation outside of the analysis range produces poor, and consequently, invalid results. You will notice that these roots also correspond to peculiar geometry. For instance, a length of 0.65 inches and case radius of 0.78 inches describe a rather large and flat ‘pancake-shaped’ solenoid.

The Actual Optimization Problem

From the previous discussion, it is evident that the local maxima, minima, and inflection points for the penalty function occur outside the analysis range. This means that the ‘humps’ corresponding to maximum and minimum cost values occur outside the analysis range. To gain a graphical perspective for what is occurring, assume that the penalty equation is a function of two independent variables and is plotted as shown in Figure 8.9.

Figure 8.9 corresponds to a plot of solenoid cost for configurations that meet the minimum force requirement. The local maxima obviously occur outside the analysis region that is outlined in the figure. This means that the optimization problem consists of

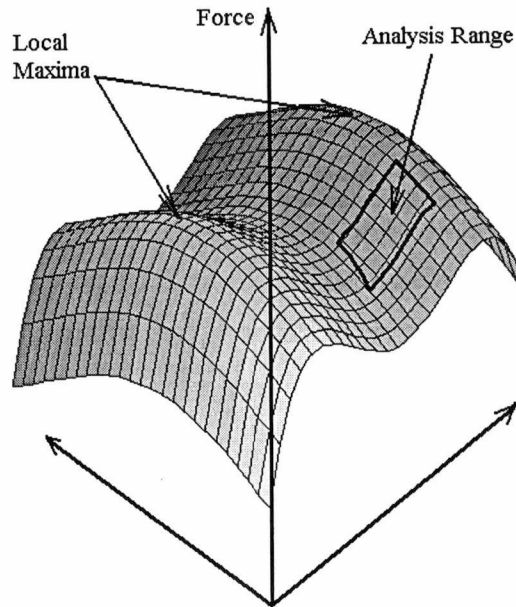


Figure 8.9. Two-Dimensional Surface Plot of a Penalty Function.

finding the lowest point on the area corresponding to the analysis region. This lowest point corresponds to the solenoid configuration with lowest cost that also satisfies the minimum force requirement.

Search Algorithm

Instead of mathematically solving for the optimal configuration, an iterative search algorithm is used. The objective of this algorithm is to iterate through all possible values of C_R , P_R , L , and W within their respective analysis ranges in search of the cost-optimal configuration. In order to perform this task, a Fortran program was developed.

Figure 8.10 is a flow diagram that summarizes the components of the program. The actual Fortran source code is given in Appendix K.

As described in the flow diagram, the required input for the program include the minimum force value, the associated stroke position, starting and stopping values for each design variable as well as the geometric step sizes for each design variable. Once these values are input by the user, nested loops are used to iterate through all possible configurations. For each iteration, force output is calculated and compared to the required minimum force value. If the configuration meets or exceeds this minimal value, solenoid cost is calculated. If the solenoid cost is less than existing minimum, the design variable values, are stored as the optimal configuration. Once the iteration process is complete, the cost-optimal configuration and its force output and cost values are displayed.

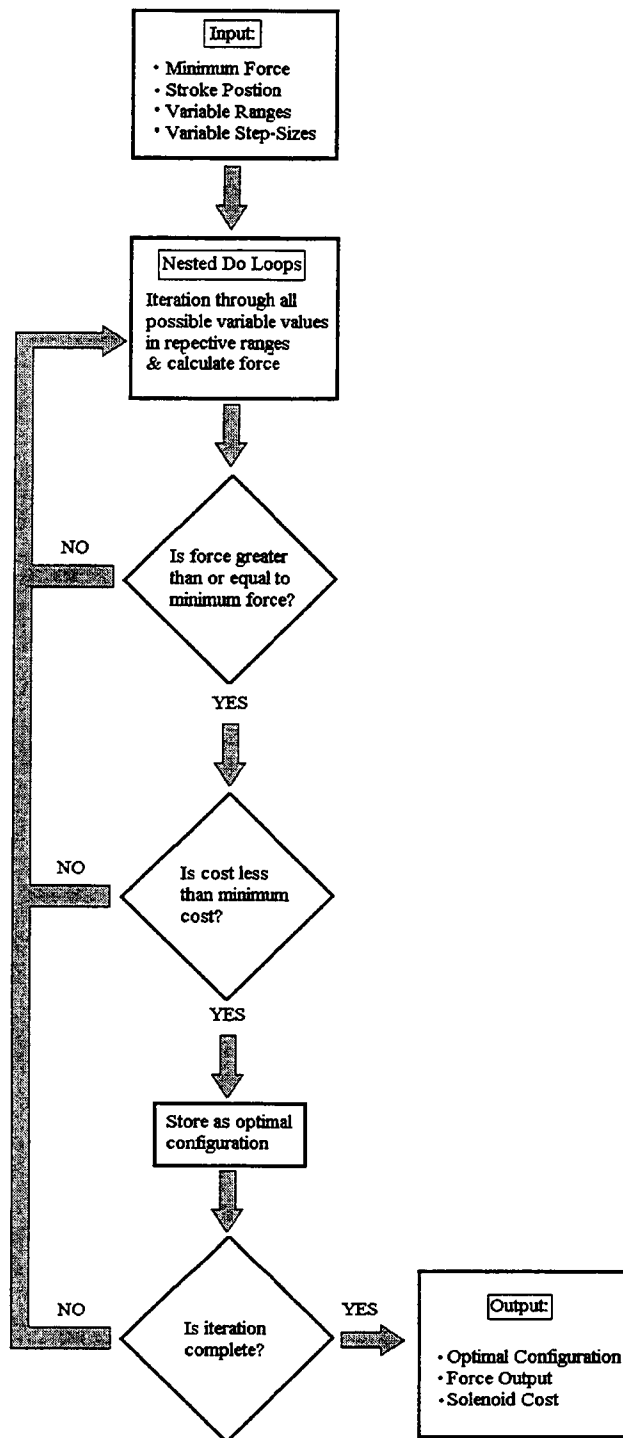


Figure 8.10. Flow Diagram for Fortran Search Algorithm.

CHAPTER IX

Design Examples

Now that the methods have been presented for determining cost-optimal solenoid designs, a number of specific design examples are presented. This chapter illustrates the steps that are carried out in order to predict cost-optimal solenoid designs for specified force and stroke position requirements.

Example 1: Minimum Force = 40.0 lb. at Stroke = 0.04 inches

Carrying out the Fortran search algorithm for these input parameters produces the output given in Figure 9.1. The force output is calculated to be 40.052 lb. and the

```
*****
Input Parameters:

Minimum Force =      40.00000
Stroke Position =      0.04000

Optimal Configuration:

Cr =      0.931000
L =      1.170000
Pr =      0.328000
W =      0.134000

Output:

Force =      40.0520
Cost =      3.1842
*****
```

Figure 9.1. Results from Design Example One.

was carried out for this configuration producing a result of 42.072 lb., which is within 5 % of the curve fitting function value.

Using the phase one Maple worksheet given in Appendix C, the optimal pole configuration for a stroke distance of 0.04 inches is determined to be,

$$A = 89^\circ$$

$$D = 0.08 \text{ inches}$$

With this, the entire cost-optimal solenoid design is defined. Table 9.1 summarizes the results from this example.

Table 9.1. Summary of Design Example One Results.

Input:
Minimum Force = 40 lb.
Stroke Position = 0.04 in.
Optimal Configuration:
$C_R = 0.931 \text{ in.}$
$L = 1.17 \text{ in.}$
$W = 0.134 \text{ in.}$
$P_R = 0.328 \text{ in.}$
$A = 89^\circ$
$D = 0.08 \text{ in.}$
Output:
Force= 40.052 lb.
Cost = \$3.18
Verification:
FEA Force = 42.072
Percent Difference = 4.8%

Example 2: Minimum Force = 25.0 lb. at Stroke = 0.07 inches

Again, the Fortran search algorithm is used to determine the optimal cost-dependent design variable values. The following Fortran output is produced for this example.

```
*****
Input Parameters:

Minimum Force =      25.00000
Stroke Position =      0.07000

Optimal Configuration:

Cr =      0.931000
L =      0.850000
Pr =      0.348000
W =      0.134000

Output:

Force =      25.0346
Cost =      2.5824
*****
```

Figure 9.2. Results from Design Example Two.

As with the previous example, a finite element analysis was carried out for this configuration which produced a force value of 23.73 lb., which is about 5% less the curve fitting function value. Recall that this is nearly the same percent difference observed in previous random tests.

After applying the phase one optimization methods to determine the optimal pole configuration, the entire cost-optimal design is defined. The results from this example, including the optimal pole configuration values, are given in Table 9.2.

Table 9.2. Summary of Design Example Two Results.

Input:
Minimum Force = 25 lb.
Stroke Position = 0.07 in.
Optimal Configuration:
$C_R = 0.931$ in.
$L = 0.85$ in.
$W = 0.134$ in.
$P_R = 0.348$ in.
$A = 30^\circ$
$D = 0.08$ in.
Output:
Force= 25.0346 lb.
Cost = \$2.58
Verification:
FEA Force = 23.73
Percent Difference = 5.2%

Example 3: Minimum Force = 30.0 lb. at Stroke = 0.11 inches

This example illustrates the output that occurs when a configuration is not found that meets or exceeds the minimum force requirement. This design requires that a minimum force of 30 lb. shall be delivered at a stroke position of 0.11 inches. Although it is possible to design a solenoid that meets these requirements, the design variable values must exceed the original finite element analysis range. When this situation

occurs, the Fortran program returns output with an appropriate message. The Fortran output for this example is given in Figure 9.3.

This problem is easily solved by increasing the finite element analysis range for each design variable. By expanding the database of finite element results in this way, a new interpolation function may be determined that is valid over a wider range of values. The Fortran search algorithm is easily modified by updating the interpolation function within the force subroutine. Also, the search limits within the program must be changed to match the range that has been analyzed.

```
*****
      Input Parameters:

      Minimum Force =      25.00000
      Stroke Position =    0.07000

      No Configurations will meet this
      Requirement For this Analysis Range!
*****
```

Figure 9.3. Results from Design Example Three.

CHAPTER X

Conclusions and Recommendations

This study has precisely outlined the steps necessary to identify the cost optimal solenoid design for a given force and stroke position requirement. As with almost any engineering analysis, the final step involves assessing the quality of the answers that are produced. In other words, the following question must be answered: *How much confidence should one have in the solenoid designs produced from this study?* While a number of trials could be carried out where solenoids are built and tested, it would still be difficult, if not impossible, to determine if a better design exists.

In order to evaluate the quality of the designs that are produce from this method, one must separately establish confidence levels for each step of the process. Unfortunately, the error that exists in each step of the optimization process has an additive effect on the final design. However, if it can be proved that the error in each step is reasonably small, the result is a high level of confidence in the answer.

The first step of the solenoid optimization process involves producing a database of finite element results. As discussed in early chapters of this report, there are a number of factors that may effect the accuracy of the finite element method. These include mesh size and input parameters such as the B-H curve data. As described in Chapter IV, these were chosen to produce results that most closely match existing empirical data, thus

minimizing the overall error. In fact, it is shown in Chapter IV, that the finite element data almost exactly matches the empirical data. In turn, it can be assumed that the finite element method can be used to almost exactly predict the force produce by any small push-pull solenoid.

The second task of the optimization process involves using the finite element database to generate force functions for phases one and two. To produce the curve fitting functions in each phase, the method of least squares is used to curve fit a polynomial to respective data set. As described in Chapter V, there are a number of factors that govern the quality of a curve fitting function. These include the amount of data that exists, as well as the form of the curve fitting function. It was also described how a high degree curve fitting function may cause instabilities and thus produce an inaccurate approximation.

In developing the curve fitting functions for phases one and two, great care was taken to ensure that the accuracy was as high as possible. The coefficient of determination (r^2 value) was used to quantitatively assess the accuracy of the force functions, while random tests were used to test for instabilities. In addition, it was shown that the optimal pole configurations predicted from the phase one force function agree with previously documented observations. Consequently, the curve fitting functions can be assumed to be accurate tools for evaluating solenoid performance.

Upon defining force functions, the next step involves using the functions to predict optimal design configurations. In phase one, simple calculus is used to find the maximum force value that exists over the original finite element analysis range. These

values are exactly determined by finding the global maxima over the analysis range. As a result, there no error associated with this step.

In phase two, the penalty method was unable to predict optimal configurations within the analysis range, therefore a Fortran search algorithm was introduced. The program is simple in that it searches all possible configurations within the analysis range in order to identify the cost-optimal configuration. Again, there is no error associated with this final step in the optimization study.

By examining the process in this way, a high level of confidence in the designs produced from this study can be justified. At the same time, it is difficult to assume that these solenoid designs are optimal. This comes from the fact that not all geometric considerations were included in the study. For instance, there are other less effecting design variables that could be included such as washer thickness, plunger cap thickness, and plunger cap diameter.

Another important consideration of this study includes the assumptions that have been made. These assumptions include:

- Constant electric current of 3.2 amps
- Steady-state system that neglects thermal effects on solenoid performance
- Magnetic permeability is the same for all steel components and corresponds to that of the experimental solenoid (see Chapter IV)
- Relative magnetic permeability of brass and nylon bobbin equal to one (same as air)

Also, the dimensions of the solenoid that are not analyzed as a design variable maintain constant values.

As explained in Chapter IX, further improvement can be made by extending the finite element analysis range. In doing so, the geometric range of possible optimal solenoid configurations is increased. Also, simply adding finite element results to the database acts to increase the accuracy of the interpolation function. Keep in mind, however, that the addition of finite element results calls for a larger computational effort. The important point is to maintain a balance between accuracy and computational cost.

As stated in the first chapter, these methods are in no way limited to the design of solenoids. The finite element method is an effective tool for analyzing and characterizing the effects of design variables on product performance. The method is especially useful in analyzing nonlinear systems such as the solenoid. Furthermore, the electromagnetic analysis of the solenoid illustrates the wide range of finite element applications that extend beyond more traditional structural and fluid flow analysis. As a result, the methods of this study have a vast range of potential applications in engineering design optimization.

List of References

List of References

1. Baker, A. J. and D. W. Pepper. *Finite Elements 1-2-3*. McGraw-Hill, Inc., New York, 1991.
2. Bhatti, M. Asghar. "Optimum Cost Design of Partially Composite Steel Beams Using LRFD," *Engineering Journal (Chicago-Ill.)*, Vol. 33, No. 1, 1996, pp. 18-29.
3. Buys, J. D. and D. G. Kroger. "Cost-Optimal Design of Cooling Towers Through Mathematical Programming Techniques," *Journal of Heat Transfer*, Vol. 111, May 1989, pp. 322-327.
4. Chatfield, C. *Statistics for Technology: A Course in Applied Statistics*, Halsted Press, New York, 1970.
5. *COSMOS/M Version 1.70 User's Guide, Vol. 1*, Structural Research and Analysis Corp., Santa Monica, CA, May 1993.
6. Haug, E. J. and J. S. Arora. *Applied Optimal Design*, John Wiley & Sons, Inc., New York, 1979.
7. Korngold, J. and G. Gabriele, J. Renaud, and G. Knott. "Applications of Multidisciplinary Design Optimization to Electronic Package Design" *Proceedings, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*. Cleveland, Ohio, Sept. 1992, pp. 1-5.
8. Korngold, J. and G. Gabriele. "Multidisciplinary Analysis and Optimization of Discrete Problems Using Response Surface Methods," *Journal of Mechanical Design*, Vol. 119, Dec. 1997 pp. 427-433.
9. Kirkpatrick, S., O. D. Gelatt Jr., and M. P. Vecchi. "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598. May 13, 1983, pp. 671-680.
10. Linnik, Y. V. *Method of Least Squares and Principles of the Theory of Observations*, Pergamon Press, New York, 1961.
11. Lucas-Ledex, *Solenoid Design Guide*, Vandalia, OH, 1991.

12. *Matlab Version 5.2 User's Guide*, The Mathworks Inc., Natick, MA, Jan. 1998.
13. Rizzoni, G. *Principles and Applications of Electrical Engineering*, Richard D. Irwin, Inc., New York, 1993.
14. Sadagopan D. and R. Pitchumani. "A combinatorial Optimization Approach Composite Materials Tailoring," *Journal of Mechanical Design*, Vol. 119, Dec. 1997, pp. 494-503.
15. Schmidt, L. C. and J. Cagan. "Optimal Configuration Design: An Integrated Approach Using Grammars," *Journal of Mechanical Design*, Vol. 120, Mar. 1998, pp. 2-7.
16. Speckhart, Dr. F. H. University of Tennessee, Personal Conversations, 1997-1998.
17. *The American College Dictionary*, Random House, New York, pp. 127.

Appendices

Appendix A

Example Problem

COSMOS Finite Element Input File

How to Perform Finite Element Analysis:

1. Type *newprob problemname*
2. Type file filename.IPT
3. Type *R_MAG* at *GEO>* prompt to perform anlysis
4. Obtain output in *problemname.OUT* file

```

C*
C*  COSMOS/M      Geostar V1.75
C*  David R. Michael
C*  Copyright 1998
C*
C*  Filename appen_A.IPT
C*
C*  Directions:  To run this program type
C*                file append_A.IPT at the
C*                GEO> prompt.  Type r_mag
C*                to perform the analysis.

```

C* Point Input Statements

```

PT,1,0,0,0,
PT,2,0,10,0,
PT,3,4,0,0,
PT,4,4,10,0,
PT,5,4.2,0,0,
PT,6,8,0,0,
PT,7,8,10,0,
PT,8,4.2,14,0,
PT,7,8,14,0,
PT,9,12,0,0,
PT,10,12,14,0,
PT,11,4.2,18,0,
PT,12,8,18,0,
PT,13,12,30,0,
PT,14,8,30,0,
PT,15,8,32,0,
PT,16,12,32,0,
PT,17,8,38,0,
PT,18,12,38,0,
PT,19,4.2,30,0,
PT,20,4.2,32,0,
PT,21,4.2,34,0,
PT,22,8,34,0,
PT,23,4.2,38,0,
PT,24,4,30,0,
PT,25,4,32,0,
PT,26,4,34,0,
PT,27,4,38,0,
PT,28,0,30,0,
PT,29,0,32,0,
PT,30,0,38,0,

```

C* Surface Input Statements

```

SF4PT,1,1,2,4,3,0,
SF4PT,2,6,7,8,5,0,
SF4PT,3,9,10,7,6,0,
SF4PT,4,3,5,19,24,0,
SF4PT,5,11,12,14,19,0,
SF4PT,6,14,15,20,19,0,
SF4PT,7,19,24,25,20,0,
SF4PT,8,24,28,29,25,0,
SF4PT,9,20,21,22,15,0,
SF4PT,10,4,2,28,24,0,
SF4PT,11,8,7,12,11,0,
SF4PT,12,7,10,13,14,0,
SF4PT,13,14,13,16,15,0,
SF4PT,14,15,16,18,17,0,
SF4PT,15,21,22,17,23,0,
SF4PT,16,26,21,23,27,0,
SF4PT,17,27,30,29,25,0,

```

SF4PT,18,25,20,21,26,0,

C* Meshing Statements

EGROUP,1,MAG2D,1,0,0,0,0,0,0,0,
MPROP,1,MPERM,1.,

M_SF,1,1,1,4,5,6,1,2,
M_SF,2,2,1,4,7,4,1,1,
M_SF,3,3,1,4,7,4,1,1,
M_SF,4,4,1,4,1,15,1,1,
M_SF,5,5,1,4,4,6,1,1,
M_SF,6,6,1,4,4,4,2,1,
M_SF,7,7,1,4,1,4,1,2,
M_SF,9,9,1,4,2,4,1,1,
M_SF,8,8,1,4,6,4,0.5,2,

C* Define B-H Curve

CURDEF,BHC,1,1,5.78053,8000.,8.04224,9500.,9.04752,10000.,11.18374,110&
00.,12.81732,11500.,16.08448,12500.,23.8754,14000.,42.7244,15500.,75.3&
96,16500,100.528,17000.,

C* Meshing Statements

MPROP,2,MPERM,1.,
M_SF,10,10,1,4,6,10,0.5,1,
M_SF,11,11,1,4,4,2,1,1,
M_SF,12,12,1,4,4,8,1,1,
M_SF,13,13,1,4,4,4,1,2,
M_SF,14,14,1,4,4,6,1,1,
M_SF,15,15,1,4,4,4,1,1,
M_SF,17,17,1,4,6,6,0.5,1,
M_SF,16,16,1,4,1,4,1,1,
M_SF,18,18,1,4,1,2,1,1,

C* Boundary Conditions

NPCR,3,0,3,1,
NPCR,12,0,12,1,
NPCR,7,0,7,1,
NPCR,10,0,10,1,
NPCR,9,0,9,1,
NPCR,43,0,43,1,
NPCR,45,0,45,1,
NPCR,48,0,48,1,
NPCR,46,0,46,1,
NPCR,49,0,49,1,
NPCR,55,0,55,1,
NPCR,57,0,57,1,
NPCR,36,0,36,1,
NPCR,1,0,1,1,
NPCR,53,0,53,1,

C* Current Denstiy

JESF,5,10.,5,1,
JESF,6,10.,6,1,
JESF,9,10.,9,1,

C* Merge Nodes

NMERGE,1,561,1,0.0001,0,1,0,

C* Analysis Set Up

EM_OUTPUT,1,0

A_MAGNETIC,S,CGS,0.0001,15,1,0,0,

Appendix B

Small Push-Pull Solenoid COSMOS Finite Element Input File

```

C*
C* COSMOS M      Geostar V1.75
C* David R. Michael
C* Copyright 1998
C*
C* Filename appen_B.IPT
C*
C* Directions:  To run this program type
C*               file append_B.IPT at the
C*               GEO> prompt.  Type r_mag
C*               to perform the analysis.

C* Point Input Statements

PT 2 0 0 0
PT 2 0.002388 0 0
PT 3 .008738 0 0
PT 4 .008738 .0127 0
PT 5 0.02054 0.0127 0
PT 6 0.02054 0.0371 0
PT 7 .009345 .0371 0
PT 8 .009345 .034941 0
PT 9 0.01797 0.034941 0
PT 10 0.01797 0.014859 0
PT 11 0.00889 0.014859 0
PT 12 0.00889 0.02337 0
PT 13 0.00888 0.02337 0
PT 14 0.00888 0.0254 0
PT 15 0.002388 0.0254 0
PT 16 0.01141 0.034179 0
PT 17 0.01145 0.034179 0
PT 18 0.01141 0.016383 0
PT 19 0.01145 0.016383 0

C* Points Corresponding to Plunger

PT 20 0.002388 0.02794 0
PT 21 .0065786 .02794 0
PT 22 .0065786 .025908 0
PT 23 .00889 .025908 0
PT 24 .00889 .039903 0
PT 25 0.01146 0.039903 0
PT 26 0.01146 0.041427 0
PT 27 .002388 .041427 0

C* Point Corresponding to Outer Edge

PT 28 0 .045 0
PT 29 .025 .045 0
PT 30 0 -.005 0
PT 31 .025 -.005 0

C* Statements to Relocate Plunger for Stroke Position
C* Datum is an offset of .1 inches (= .00254 m)

C* PTRELOC 22,27,1,0,0,0,.000508,0.0
C* PTRELOC 20,21,1,0,0,0,.000508,0.0

```

C* Curve Input Statements

CRLINE 1 2 3
CRLINE 2 3 4
CRLINE 3 4 5
CRLINE 4 5 6
CRLINE 5 6 7
CRLINE 6 7 8
CRLINE 7 8 9
CRLINE 8 9 10
CRLINE 9 10 11
CRLINE 10 11 12
CRLINE 11 12 13
CRLINE 12 13 14
CRLINE 13 14 15
CRLINE 14 15 2
CRLINE 15 16 17
CRLINE 16 17 19
CRLINE 17 19 18
CRLINE 18 18 16
CRLINE 19 20 21
CRLINE 20 21 22
CRLINE 21 22 23
CRLINE 22 23 24
CRLINE 23 24 25
CRLINE 24 25 26
CRLINE 25 26 27
CRLINE 26 27 20
CRLINE 27 28 29
CRLINE 28 29 31
CRLINE 29 31 30
CRLINE 30 30 28

C* Contour Definition Statements

CT 1 0 .001 1 26 0
CT 2 0 .001 1 6 0
CT 3 0 .001 1 18 0
CT 4 0 .001 1 30 0

C* Region Definition Statements

RG 1 1 1 0
RG 2 1 2 0
RG 3 4 4 2 1 3 0
RG 4 1 3 0

C* Establish the Element Size (Edge Length = .001m)

RGDENSITY 1 4 1 .001 0.0001 1 0 1.0

C* Element Group and Meshing of Air and Copper

EGROUP 1 MAG2D 1 0 0 0 0 0 0
MPROP 1 MPERM .1256637E-5
MA_RG 3 3 1 3 1 0
MA_RG 4 4 1 3 1 0

C* Nonlinear Input (B-H curve data)

CURDEF BHC 1 1 410 .66 590 .79 670 .83 840 .91 970 .95 1230 1.04 1850 &
1.16 3350 1.25 5950 1.37

C* Properties for Iron Core and Meshing of Core

MPROP 2 MPERM 1
MA_RG 2 2 1 3 1 0

C* Element Group, Properties and Meshing for plunger (must be movable)

EGROUP 2 MAG2D 1 1 0 0 0 0 0
ACTSET EG 2
MA_RG 1 1 1 3 1 0

C* Merge nodes

NMERGE 1 5000 1 0.0001 0 0 0

C* Establish zero boundary conditions at edges

NPCR 27 0 30 1

C* Input the current density of copper coil

JERG 4 14.9265E6 4 1

C* Analysis Setup

EM_OUTPUT 1 1 1 0 0 0
A_MAGNETIC S MIS 0.0001 15 1 0 0

Appendix C

Low Reluctance Model
COSMOS Finite Element Input File

C*
 C* COSMOS/M Geostar V1.75
 C* David R. Michael
 C* Copyright 1998
 C*
 C* Filename appen_C.IPT
 C*
 C* Directions: To run this program type
 C* file append_C.IPT at the
 C* GEO> prompt. Type r_mag
 C* to perform the analysis.

C* Point Input Statements

PT 2 0 0 0
 PT 2 0.002388 0 0
 PT 3 .008738 0 0
 PT 4 .008738 .001 0
 PT 5 .025 0.001 0
 PT 6 .025 0.041469 0
 PT 7 .009346 .041469 0
 PT 8 .009346 .034941 0
 PT 9 0.01797 0.034941 0
 PT 10 0.01797 0.014859 0
 PT 11 0.00889 0.014859 0
 PT 12 0.00889 0.02337 0
 PT 13 0.00588 0.02337 0
 PT 14 0.00470677 0.0254 0
 PT 15 0.002388 0.0254 0
 PT 16 0.01041 0.034179 0
 PT 17 0.01645 0.034179 0
 PT 18 0.01041 0.016383 0
 PT 19 0.01645 0.016383 0

C* Points Corresponding to Plunger

PT 20 0.002388 0.02794 0
 PT 21 .00540537 .02794 0
 PT 22 .0065786 .025908 0
 PT 23 .00889 .025908 0
 PT 24 .00889 .044272 0
 PT 25 0.01746 0.044272 0
 PT 26 0.01746 .045796 0
 PT 27 .002388 .045796 0

C* Points Corresponding to Outer Edge

PT 28 0 .049369 0
 PT 29 .03 .049369 0
 PT 30 0 -.005 0
 PT 31 .03 -.005 0

C* Statements to Relocate Plunger for Strike Position
 C* Datum is an offset of .1 inches (= .00254 m)

C* PTRELOC,22,27,1,0,0,0,-.000508,0,0
 C* PTRELOC,20,21,1,0,0,0,-.000508,0,0

C* Curve Input Statements

CRLINE 1 2 3
 CRLINE 2 3 4
 CRLINE 3 4 5
 CRLINE 4 5 6

```

CRLINE 5 6 7
CRLINE 6 7 8
CRLINE 7 8 9
CRLINE 8 9 10
CRLINE 9 10 11
CRLINE 10 11 12
CRLINE 11 12 13
CRLINE 12 13 14
CRLINE 13 14 15
CRLINE 14 15 2
CRLINE 15 16 17
CRLINE 16 17 19
CRLINE 17 19 18
CRLINE 18 18 16
CRLINE 19 20 21
CRLINE 20 21 22
CRLINE 21 22 23
CRLINE 22 23 24
CRLINE 23 24 25
CRLINE 24 25 26
CRLINE 25 26 27
CRLINE 26 27 20
CRLINE 27 28 29
CRLINE 28 29 31
CRLINE 29 31 30
CRLINE 30 30 28

```

C* Contour Definition Statements

```

CT 1 0 .001 1 26 0
CT 2 0 .001 1 6 0
CT 3 0 .001 1 18 0
CT 4 0 .001 1 30 0

```

C* Region Definition Statements

```

RG 1 1 1 0
RG 2 1 2 0
RG 3 4 4 2 1 3 0
RG 4 1 3 0

```

C* Establish the element size (Size = .001)

```

RGDENSITY 1 4 1 .001 0.0001 1 0 1.0

```

C* Element Group and Meshing of Air and Copper

```

EGROUP 1 MAG2D 1 0 0 0 0 0 0 0
MPROP 1 MPERM .1256637E-5
MA_RG 3 3 1 3 0 0
MA_RG 4 4 1 3 0 0

```

C* Nonlinear Input (B-H curve data)

```

CURDEF BHC 1 1 410 .66 590 .79 670 .83 840 .91 970 .95 1230 1.04 1850 &
1.16 3350 1.29 5950 1.37

```

C* Properties for Iron Core and Meshing of Core

MPROP 2 MPERM 1
MA_RG 2 2 1 3 0 0

C* Element Group, Properties and Meshing for plunger (must be movable)

EGROUP 2 MAG2D 1 1 0 0 0 0 0 0
ACTSET EG 2
MA_RG 1 1 1 3 0 0

C* Merge nodes

NMERGE 1 5000 1 0.0001 0 0 0

C* Establish zero boundary conditions at edges

NPCR 27 0 30 1

C* Input the current density of copper coil

JERG 4 12.8e6 4 1

C* Analysis Setup

EM_OUTPUT 1 1 1 0 0 0
A_MAGNETIC S MKS 0.0001 15 1 0 0

Appendix D

Phase One Finite Element Results

Descriptors		
Notch Distance (in.)	Notch Angle (deg)	Stroke Distance (in.)
D1=0.02	A1=110	S1=0.04
D2=0.04	A2=90	S2=0.06
D3=0.06	A3=75	S3=0.08
D4=0.08	A4=60	S4=0.11
D5=0.10	A5=45	S5=0.14
D6=0.12	A6=30	
D7=0.14	A7=25	

Parameters	Force (lb)	Normalized Force
D4 S1 A3	29.532	1.881
D4 S2 A3	20.394	1.299
D4 S3 A3	14.083	0.897
D4 S4 A3	8.635	0.550
D4 S5 A3	5.872	0.374
Average=	15.703	1.000
D1 S1 A3	27.789	1.770
D1 S2 A3	18.997	1.210
D1 S3 A3	13.267	0.845
D1 S4 A3	7.819	0.498
D1 S5 A3	5.212	0.332
D2 S1 A3	29.045	1.850
D2 S2 A3	19.625	1.250
D2 S3 A3	13.471	0.858
D2 S4 A3	8.038	0.512
D2 S5 A3	5.369	0.342
D3 S1 A3	29.328	1.868
D3 S2 A3	20.096	1.280
D3 S3 A3	13.879	0.884
D3 S4 A3	8.258	0.526
D3 S5 A3	5.589	0.356
D5 S1 A3	28.354	1.806
D5 S2 A3	19.892	1.267
D5 S3 A3	14.224	0.906
D5 S4 A3	8.619	0.549
D5 S5 A3	5.762	0.367
D6 S1 A3	27.475	1.750
D6 S2 A3	19.688	1.254
D6 S3 A3	14.114	0.899
D6 S4 A3	8.478	0.540
D6 S5 A3	5.636	0.359
D7 S1 A3	26.470	1.686
D7 S2 A3	19.060	1.214
D7 S3 A3	14.004	0.892
D7 S4 A3	8.352	0.532
D7 S5 A3	5.511	0.351
D4 S1 A1	23.613	1.504

D4	S2	A1	15.841	1.009
D4	S3	A1	10.676	0.680
D4	S4	A1	5.338	0.340
D4	S5	A1	3.454	0.220
D4	S1	A2	32.185	2.050
D4	S2	A2	20.771	1.323
D4	S3	A2	13.769	0.877
D4	S4	A2	8.243	0.525
D4	S5	A2	5.511	0.351
D4	S1	A3	30.144	1.920
D4	S2	A3	20.960	1.335
D4	S3	A3	14.036	0.894
D4	S4	A3	8.227	0.524
D4	S5	A3	5.558	0.354
D4	S1	A5	29.045	1.850
D4	S2	A5	21.085	1.343
D4	S3	A5	14.915	0.950
D4	S4	A5	9.137	0.582
D4	S5	A5	6.107	0.389
D4	S5	A6	27.632	1.760
D4	S5	A6	21.195	1.350
D4	S5	A6	15.386	0.980
D4	S5	A6	9.577	0.610
D4	S5	A7	6.280	0.400
D4	S5	A7	26.847	1.710
D4	S5	A7	20.253	1.290
D4	S5	A7	14.287	0.910
D4	S5	A7	9.232	0.588
D4	S5	A7	6.154	0.392

Appendix E
Phase Two Data

Force (lb.)	S (in.)	C _R (in.)	P _R (in.)	W (in.)	L (in.)
25.388	0.04	0.813	0.350	0.105	0.962
20.856	0.06	0.813	0.350	0.105	0.962
15.795	0.08	0.813	0.350	0.105	0.962
10.988	0.11	0.813	0.350	0.105	0.962
11.042	0.04	0.691	0.350	0.105	0.962
7.492	0.06	0.691	0.350	0.105	0.962
4.709	0.08	0.691	0.350	0.105	0.962
2.882	0.11	0.691	0.350	0.105	0.962
16.244	0.04	0.731	0.350	0.105	0.962
12.216	0.06	0.731	0.350	0.105	0.962
8.263	0.08	0.731	0.350	0.105	0.962
5.241	0.11	0.731	0.350	0.105	0.962
21.074	0.04	0.772	0.350	0.105	0.962
16.664	0.06	0.772	0.350	0.105	0.962
12.047	0.08	0.772	0.350	0.105	0.962
7.995	0.11	0.772	0.350	0.105	0.962
29.110	0.04	0.853	0.350	0.105	0.962
24.741	0.06	0.853	0.350	0.105	0.962
19.448	0.08	0.853	0.350	0.105	0.962
14.040	0.11	0.853	0.350	0.105	0.962
32.470	0.04	0.894	0.350	0.105	0.962
28.112	0.06	0.894	0.350	0.105	0.962
22.855	0.08	0.894	0.350	0.105	0.962
17.058	0.11	0.894	0.350	0.105	0.962
35.578	0.04	0.934	0.350	0.105	0.962
31.182	0.06	0.934	0.350	0.105	0.962
25.948	0.08	0.934	0.350	0.105	0.962
20.060	0.11	0.934	0.350	0.105	0.962
23.468	0.04	0.813	0.298	0.105	0.962
19.243	0.06	0.813	0.298	0.105	0.962
16.422	0.08	0.813	0.298	0.105	0.962
11.952	0.11	0.813	0.298	0.105	0.962
24.496	0.04	0.813	0.315	0.105	0.962
20.890	0.06	0.813	0.315	0.105	0.962
16.436	0.08	0.813	0.315	0.105	0.962
11.886	0.11	0.813	0.315	0.105	0.962
25.333	0.04	0.813	0.333	0.105	0.962
20.465	0.06	0.813	0.333	0.105	0.962
16.760	0.08	0.813	0.333	0.105	0.962

11.515	0.11	0.813	0.333	0.105	0.962
24.970	0.04	0.813	0.368	0.105	0.962
19.355	0.06	0.813	0.368	0.105	0.962
15.398	0.08	0.813	0.368	0.105	0.962
10.195	0.11	0.813	0.368	0.105	0.962
23.766	0.04	0.813	0.385	0.105	0.962
18.147	0.06	0.813	0.385	0.105	0.962
13.949	0.08	0.813	0.385	0.105	0.962
9.335	0.11	0.813	0.385	0.105	0.962
22.390	0.04	0.813	0.403	0.105	0.962
16.874	0.06	0.813	0.403	0.105	0.962
13.003	0.08	0.813	0.403	0.105	0.962
8.417	0.11	0.813	0.403	0.105	0.962
23.368	0.04	0.813	0.350	0.074	0.962
19.882	0.06	0.813	0.350	0.074	0.962
15.363	0.08	0.813	0.350	0.074	0.962
10.792	0.11	0.813	0.350	0.074	0.962
24.768	0.04	0.813	0.350	0.084	0.962
20.538	0.06	0.813	0.350	0.084	0.962
15.627	0.08	0.813	0.350	0.084	0.962
10.903	0.11	0.813	0.350	0.084	0.962
25.214	0.04	0.813	0.350	0.095	0.962
20.753	0.06	0.813	0.350	0.095	0.962
15.724	0.08	0.813	0.350	0.095	0.962
10.952	0.11	0.813	0.350	0.095	0.962
25.509	0.04	0.813	0.350	0.116	0.962
20.947	0.06	0.813	0.350	0.116	0.962
15.846	0.08	0.813	0.350	0.116	0.962
11.014	0.11	0.813	0.350	0.116	0.962
25.563	0.04	0.813	0.350	0.126	0.962
20.990	0.06	0.813	0.350	0.126	0.962
15.876	0.08	0.813	0.350	0.126	0.962
11.025	0.11	0.813	0.350	0.126	0.962
25.580	0.04	0.813	0.350	0.137	0.962
21.010	0.06	0.813	0.350	0.137	0.962
15.888	0.08	0.813	0.350	0.137	0.962
11.033	0.11	0.813	0.350	0.137	0.962
14.361	0.04	0.813	0.350	0.105	0.673
10.735	0.06	0.813	0.350	0.105	0.673
6.877	0.08	0.813	0.350	0.105	0.673
2.847	0.11	0.813	0.350	0.105	0.673

18.353	0.04	0.813	0.350	0.105	0.770
14.318	0.06	0.813	0.350	0.105	0.770
9.852	0.08	0.813	0.350	0.105	0.770
4.309	0.11	0.813	0.350	0.105	0.770
22.139	0.04	0.813	0.350	0.105	0.866
17.712	0.06	0.813	0.350	0.105	0.866
12.881	0.08	0.813	0.350	0.105	0.866
5.983	0.11	0.813	0.350	0.105	0.866
27.629	0.04	0.813	0.350	0.105	1.058
23.940	0.06	0.813	0.350	0.105	1.058
18.281	0.08	0.813	0.350	0.105	1.058
9.696	0.11	0.813	0.350	0.105	1.058
30.213	0.04	0.813	0.350	0.105	1.154
26.252	0.06	0.813	0.350	0.105	1.154
20.896	0.08	0.813	0.350	0.105	1.154
11.471	0.11	0.813	0.350	0.105	1.154
32.319	0.04	0.813	0.350	0.105	1.251
28.151	0.06	0.813	0.350	0.105	1.251
23.280	0.08	0.813	0.350	0.105	1.251
13.319	0.11	0.813	0.350	0.105	1.251
17.309	0.04	0.813	0.350	0.105	0.962
13.278	0.06	0.813	0.350	0.105	0.962
9.170	0.08	0.813	0.350	0.105	0.962
5.895	0.11	0.813	0.350	0.105	0.962
20.173	0.04	0.813	0.350	0.105	0.962
15.854	0.06	0.813	0.350	0.105	0.962
11.390	0.08	0.813	0.350	0.105	0.962
7.515	0.11	0.813	0.350	0.105	0.962
22.890	0.04	0.813	0.350	0.105	0.962
18.377	0.06	0.813	0.350	0.105	0.962
13.611	0.08	0.813	0.350	0.105	0.962
9.226	0.11	0.813	0.350	0.105	0.962
27.708	0.04	0.813	0.350	0.105	0.962
23.238	0.06	0.813	0.350	0.105	0.962
17.970	0.08	0.813	0.350	0.105	0.962
12.776	0.11	0.813	0.350	0.105	0.962
29.928	0.04	0.813	0.350	0.105	0.962
25.470	0.06	0.813	0.350	0.105	0.962
20.119	0.08	0.813	0.350	0.105	0.962
14.577	0.11	0.813	0.350	0.105	0.962
32.072	0.04	0.813	0.350	0.105	0.962

27.563	0.06	0.813	0.350	0.105	0.962
22.217	0.08	0.813	0.350	0.105	0.962
16.401	0.11	0.813	0.350	0.105	0.962

Appendix F

Maple Worksheet

Example Least Squares Problem

Appendix F: Least Square Example Problem

Worksheet Objective: To interpolate a polynomial to a given data set by the method of least squares.

Thesis: "Design of a Cost-Optimal Solenoid"

By: David R. Michael

University of Tennessee, Knoxville

Copyright August 1998: All Rights Reserved.

Filename: appen_F.mws

Insert Restart statement to clear memory

> restart;

Input X and Y Data Arrays

> X:=array(1..6,[0,0.3,0.8,1.1,1.6,2.3]):
> Y:=array(1..6,[0.5,0.82,1.14,1.25,1.35,1.4]):

Least Squares Summation and Calculation of Coefficients

> errsq:=sum('(Y[k]-c1-c2*X[k]-c3*X[k]^2)^2','k'=1..6):
> eq1:=diff(errsq,c1)=0:
> eq2:=diff(errsq,c2)=0:
> eq3:=diff(errsq,c3)=0:
> coefs:=solve({eq1,eq2,eq3},{c1,c2,c3});
coefs:={c3=-.2387054047,c2=.9190628219,c1=.5317737830}

Polynomial Determination

```

*****
> poly:=(c1+c2*x-c3*x^2):
> fcn:=subs(coefs,poly);
      fcn := .5317737830 + .9190628219 x - .2387054047 x^2
*****
Calculation of Ybar, Yhat, explained variation, and total variation
*****
> ybar:=sum(Y[k], 'k=1..6')/6:
> for i from 1 to 6 do yha[i]:=c1+c2*X[i]+c3*X[i]^2 od:
> for i from 1 to 6 do yhat[i]:=subs(coefs,yha[i]) od:
> expvar:=sum((yhat[k]-ybar)^2, 'k=1..6'):
> totvar:=sum((Y[k]-ybar)^2, 'k=1..6'):
*****
Calculation of Coefficient of Determination (rsq)
*****
> rsq:=expvar/totvar;
      rsq := .9920991637

```

Appendix G

Maple Worksheet
Phase One Calculations

Appendix G: Phase One Maple Worksheet

Worksheet Objective: To develop the phase one interpolation force function; To compute its coefficient of determination; To compute interpolation function values for random testing;

Thesis: "Design of a Cost-Optimal Solenoid"

By: David R. Michael

University of Tennessee, Knoxville

Copyright August 1998: All Rights Reserved

Filename: appen_G.mws

Insert Restart statement to clear memory and enable plots

```
> restart;
```

```
> with(plots):
```

Input Data Arrays for F, S, A, and D.

Note: Since D is a reserved character in Maple, H is used instead

```
> f:=array(1..65,[1.881,1.299,.897,.55,.374,1.77,1.21,.845,.498,.332  
    ,1.85,1.25,.858,.512,.342,1.868,1.28,.884,.526,.356,1.806,1.267,.9  
    06,.549,.367,1.75,1.254,.899,.54,.359,1.686,1.214,.892,.532,.351,1.  
    .5,1.,.68,.34,.22,2.05,1.323,.877,.525,.351,1.92,1.335,.894,.524,.  
    354,1.85,1.343,.95,.582,.389,1.76,1.35,.98,.61,.4,1.71,1.29,.91,.5  
    88,.392])):  
  
> h:=array(1..65,[1,1,1,1,1,.25,.25,.25,.25,.25,.5,.5,.5,.5,.5,.75,.  
    75,.75,.75,1.25,1.25,1.25,1.25,1.25,1.5,1.5,1.5,1.5,1.5,1.75,1.  
    75,1.75,1.75,1.75,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
    1,1,1,1,1])):  
  
> s:=array(1..65,[.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5,.75,1,1.  
    375,1.75,.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.  
    75,.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5  
    ,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5,.75,1,1.375,1.75,.5,.75,1  
    ,1.375,1.75])):  
  
> a:=array(1..65,[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
    1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
```

```
1,1,1,1,1,1,1,1,1,1,1,1.83,1.83,1.83,1.83,1.83,1.5,1.5,1.5,1.5,1.5,1
.25,1.25,1.25,1.25,1.25,.75,.75,.75,.75,.75,.5,.5,.5,.5,.5,.4,.4,.
4,.4,.4]);
```

```
*****
Carry out Least Squares Summation and Solve for Unknown Coefficients
*****
```

```
> errsq:=sum('(f[k]-c1-c2*s[k]-c3*s[k]^2-c4*s[k]*a[k]-c5*s[k]^3-c6*s
[k]^4-c7*h[k]-c8*h[k]^2-c9*h[k]*s[k]-c10*h[k]^3-c11*h[k]^4-c12*a[k
]-c13*a[k]^2-c14*a[k]^3-c15*a[k]^4)^2','k'=1..65):
> eq1:=diff(errsq,c1)=0:
> eq2:=diff(errsq,c2)=0:
> eq3:=diff(errsq,c3)=0:
> eq4:=diff(errsq,c4)=0:
> eq5:=diff(errsq,c5)=0:
> eq6:=diff(errsq,c6)=0:
> eq7:=diff(errsq,c7)=0:
> eq8:=diff(errsq,c8)=0:
> eq9:=diff(errsq,c9)=0:
> eq10:=diff(errsq,c10)=0:
> eq11:=diff(errsq,c11)=0:
> eq12:=diff(errsq,c12)=0:
> eq13:=diff(errsq,c13)=0:
> eq14:=diff(errsq,c14)=0:
> eq15:=diff(errsq,c15)=0:

> coefs:=solve({eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq9,eq10,eq11,eq12,e
q13,eq14,eq15},{c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15
});
coefs:={c2=-4.541133317,c10=-.3227623500,c4=-.04173963955,c13=-7.501526830,
c9=.05758432523,c11=.08504618111,c7=-.04668451108,c12=4.579228500,
c1=2.562189165,c3=2.448849738,c15=-1.251120609,c14=5.146146004,c8=.3029306955
c6=.07050230309,c5=-.6369285724}
```

```
*****
Determine Interpolation Polynomial
*****
```

```

> poly:=(c1+c2*s+c3*s^2+c4*s*a+c5*s^3+c6*s^4+c7*h+c8*h^2+c9*h*s+c10*
h^3+c11*h^4+c12*a+c13*a^2+c14*a^3+c15*a^4):
>
> forcefcn:=subs(coefs,poly);
forcefcn:=2.562189165 - 4.541133317 s + 2.448849738 s^2 - .04173963955 s a
- .6369285724 s^3 + .07050230309 s^4 - .04668451108 h + .3029306955 h^2 + .05758432523 h s
- .3227623500 h^3 + .08504618111 h^4 + 4.579228500 a - 7.501526830 a^2 + 5.146146004 a^3
- 1.251120609 a^4

```

Statement For Random Testing

```

> randomtest:=15.7*2*3.1415/4.448*(subs(s=1.625,h=1.375,a=1.333,forc
efcn));
randomtest:=9.608256222
> for i from 1 to 65 do
mathforce[i]:=eval(forcefcn,[h=h[i],s=s[i],a=a[i]]) od:
>
> for i from 1 to 65 do error[i]:=((abs(f[i]-mathforce[i]))^2) od:
>
> toterror:=sum('error[k]', 'k'=1..65);
toterror:=.1211716008
>

```

Calculations for finding maxima and minima

Substitute Appropriate Stroke Position

```

> optforce:=subs(s=1.5,forcefcn);
optforce:=-.5323149226 + 4.516619041 a + .03969197677 h + .3029306955 h^2
- .3227623500 h^3 + .08504618111 h^4 - 7.501526830 a^2 + 5.146146004 a^3 - 1.251120609 a^4

```

Differentiate with respect to design variables

```

> opteq1:=diff(optforce,a)=0;
opteq1:=4.516619041 - 15.00305366 a + 15.43843801 a^2 - 5.004482436 a^3 = 0
> opteq2:=diff(optforce,h)=0;

```

$$\text{opteq2} := .03969197677 + .6058613910 h - .9682870500 h^2 + .3401847244 h^3 = 0$$

Compute Second Derivatives

```
> opteq3:=diff(optforce,a,a);
      opteq3 := -15.00305366 + 30.87687602 a - 15.01344731 a^2
> opteq4:=diff(optforce,h,h);
      opteq4 := .6058613910 - 1.936574100 h + 1.020554173 h^2
> opteq5:=diff(optforce,a,h);
      opteq5 := 0
> opteq6:=diff(optforce,h,a);
      opteq6 := 0
```

Solve for Roots

```
> optsol:=solve({opteq1,opteq2},{a,h});
optsol := {a = .5956314813, h = -.05969807799}, {a = .5956314813, h = 1.057016024},
  {a = .5956314813, h = 1.849038697}, {h = -.05969807799, a = 1.060475492},
  {h = 1.057016024, a = 1.060475492}, {h = 1.849038697, a = 1.060475492},
  {h = -.05969807799, a = 1.428815035}, {h = 1.057016024, a = 1.428815035},
  {h = 1.849038697, a = 1.428815035}
>
> for i from 1 to 9 do force[i]:=subs(optsol[i],optforce) od:
>
> print(force);
table([
  1 = .4253272819
  2 = .5319489479
  3 = .4893364989
  4 = .374996725
  5 = .481618391
  6 = .439005942
  7 = .402049193
  8 = .508670859
  9 = .466058410
])
> print(optsol[8],force=force[8]);
      {h = 1.057016024, a = 1.428815035}, force = .508670859
```

Substitute Roots into Second Derivative Equations to Check for Maxima or Minima

```
> for i from 1 to 9 do ca[i]:=subs(optsol[i],opteq3) od;
      ca1 := -1.938237977
      ca2 := -1.938237977
      ca3 := -1.938237977
      ca4 := .85686964
      ca5 := .85686964
      ca6 := .85686964
      ca7 := -1.53584788
      ca8 := -1.53584788
      ca9 := -1.53584788
> for i from 1 to 9 do cb[i]:=subs(optsol[i],opteq4) od;
      cb1 := .7251082554
      cb2 := -.300880763
      cb3 := .514278612
      cb4 := .7251082554
      cb5 := -.300880763
      cb6 := .514278612
      cb7 := .7251082554
      cb8 := -.300880763
      cb9 := .514278612
```

Print roots associated with local maxima

```
> for i from 1 to 9 do if ca[i] < 0 and cb[i] < 0 then
  print(localmax[i]=optsol[i]) fi od;
      localmax2 = { a = .5956314813, h = 1.057016024 }
      localmax8 = { h = 1.057016024, a = 1.428815035 }
```

Print Forces Created by Roots

```
> for i from 1 to 9 do rforce[i]:=force[i]*15.7 od;
> print(rforce);
```

>

Appendix H

Maple Worksheet
Cost Equations

Appendix H: Maple Worksheet for Cost Equations

Worksheet Objective: To compute the equations for solenoid cost.

Thesis: "Design of a Cost-Optimal Solenoid"

By: David R. Michael

University of Tennessee, Knoxville

Copyright August 1998: All Rights Reserved.

Filename: appen_H.mws

Insert Restart statement to clear memory

> restart;

Component 1: Washer Cost

Note: washer volume does not account for hole

```
> washercost:=0.02+0.3*(washervolume);
                                washercost := .02 + .3 washervolume
> washercost := .2e-1+.3*washervolume;
                                washercost := .02 + .3 washervolume
> washervolume:=Pi*CR^2*0.083;
                                washervolume := .083 π CR2
> totalwashercost:=simplify(expand(washercost));
                                totalwashercost := .02000000000 + .07822565708 CR2
```

Component 2. Case Cost

```
> casecost:=.36*(casevolume)+0.02*CR+0.26;
                                casecost := .36 casevolume + .02 CR + .26
> casevolume:=Pi*(CR^2-(CR-W)^2)*(L+0.15);
                                casevolume := π (CR2 - (CR - W)2) (L + .15)
> totalcasecost:=simplify(expand(casecost));
totalcasecost := 2.261946711 CR W L + .3392920066 CR W - 1.130973355 W2 L
                                - .1696460033 W2 + .02000000000 CR + .26000000000
```

Component 3. Plunger/Core Cost (plucr Cost)

```

> plucrcost:=0.25*plucrvolume+0.1*PR+0.05;
      plucrcost := .25 plucrvolume + .1 PR + .05
> plucrvolume:=Pi*PR^2*(L+.15);
      plucrvolume := π PR² (L+.15)
> totalplucrcost:=simplify(expand(plucrcost));
      totalplucrcost := .7853981635 PR² L + .1178097245 PR² + .1000000000 PR + .0500000000
Component 4. Copper Coil Cost (coilcost)
> coilcost:=0.78+0.9*(coilvolume);
      coilcost := .78 + .9 coilvolume
> coilvolume:=(L-.166)*Pi*((CR-W)^2-PR^2);
      coilvolume := (L-.166) π ((CR-W)² - PR²)
> totalcoilcost:=simplify(expand(coilcost));
totalcoilcost := .7800000000 + 2.827433389 L CR² - 5.654866777 CR W L + 2.827433389 W² L
      - 2.827433389 PR² L - .4693539425 CR² + .9387078850 CR W - .4693539425 W²
      + .4693539425 PR²
Compute the total solenoid cost (solcost)
> solcost:=simplify(expand(totalwashercost+totalcasecost+totalplucrcost+totalcoilcost));
solcost := 1.110000000 - .3911282854 CR² - 3.392920066 CR W L + 1.277999892 CR W
      + 1.696460034 W² L - .6389999458 W² + .02000000000 CR - 2.042035226 PR² L
      + .5871636670 PR² + .1000000000 PR + 2.827433389 L CR²
> costcheck:=subs(CR=.931,W=.134,L=1.1,PR=.338,solcost);
      costcheck := 3.045506904

```

Appendix I

Maple Worksheet

Altered Curve Fit Function Calculation

Appendix I: Special Force Function (Phase Two) Maple Worksheet

Worksheet Objective: To develop the phase two interpolation force function that contains quadratic terms in S, W, and PR only; To compute its coefficient of determination;

Thesis: "Design of a Cast-Optimal Solenoid"

By: David R. Michael

University of Tennessee, Knoxville

Copyright August 1998. All Rights Reserved.

Filename: appen_1.mws

Insert Restart statement to clear memory

```
> restart;
```

Input Finite Element Results as Data Arrays:

Notes: Force values are normalized by dividing actual result in (lb.) by 12.925.

Stroke Position values are normalized by dividing actual value in (inches) by 0.08.

All other data is not normalized and has units of inches

```
> F:=array(1..104,[1.391,1.142,.865,.602,1.285,1.054,.899,.655,1.342,
,1.144,.9,.651,1.388,1.121,.918,.631,1.368,1.06,.843,.558,1.302,.9
94,.764,.511,1.226,.924,.712,.461,1.28,1.089,.841,.591,1.357,1.125
,.856,.597,1.381,1.137,.861,.6,1.397,1.147,.868,.603,1.4,1.15,.87,
.604,1.401,1.151,.87,.604,.889,.883,.896,.896,.739,.688,.902,.903,
.922,.849,.77,.718,1.005,.784,.54,.236,1.213,.97,.706,.328,1.513,1
.311,1.001,.531,1.655,1.438,1.145,.628,1.77,1.542,1.275,.73,.89,.6
69,.453,.287,1.154,.913,.66,.438,1.594,1.355,1.765,.769,1.778,1.54
,1.252,.934,1.949,1.708,1.421,1.099]):
```

```
> S:=array(1..104,[S,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.  
75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,  
.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.  
5,.75,1,1.375,1,1,1,1,1,1,1,1,1,1,.5,.75,1,1.375,.5,.75,1,1.37  
5,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,
```

```

75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375]):
> PR:=array(1..104,[.35,.35,.35,.35,.298,.298,.298,.298,.315,.315,.3
15,.315,.333,.333,.333,.333,.368,.368,.368,.368,.385,.385,.385,.38
5,.403,.403,.403,.403,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.298,.315,.333,
.368,.385,.403,.298,.315,.333,.368,.385,.403,.35,.35,.35,.35,.35,
.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
5,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
.35,.35]):
> W:=array(1..104,[.105,.105,.105,.105,.105,.105,.105,.105,.105,.105
,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
.105,.105,.105,.105,.105,.074,.074,.074,.074,.084,.084,.084,.084,.
095,.095,.095,.095,.116,.116,.116,.116,.126,.126,.126,.126,.137,.1
37,.137,.137,.074,.074,.074,.074,.074,.074,.137,.137,.137,.137,.13
7,.137,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105
,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
105,.105]):
> L:=array(1..104,[.962,.962,.962,.962,.962,.962,.962,.962,.962,.962
,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,
.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.
962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.9
62,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.96
2,.962,.77,.77,.77,.77,.866,.866,.866,.866,1.058,1.058,1.058,1.058,
1.154,1.154,1.154,1.154,1.251,1.251,1.251,1.251,.962,.962,.962,.9
62,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.96
2,.962,.962,.962]):
> CR:=array(1..104,[.818,.813,.813,.813,.818,.813,.813,.813,.818,.81
3,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,
.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,
.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.
813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.8
13,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.81
3,.818,.813,.813,.813,.818,.813,.813,.813,.813,.731,.731,.731,.731,
.772,.772,.772,.853,.853,.853,.853,.894,.894,.894,.894,.934,.934,
.934,.934]):

```

```

*****
Carry out least squares summation and solve for each unknown coefficient;
*****

```

```

> errsq:=sum(' (F[k]-1.*c1-1.*c2*S[k]-1.*c3*S[k]^2-1.*c4*CR[k]-1.*c5*
W[k]-1.*c6*W[k]^2-c7*L[k]-c8*PR[k]-c9*PR[k]^2)^2', 'k'=1..104):
> eq1:=diff(errsq,c1)=0:

```

```

> eq2:=diff(errsq,c2)=0:
> eq3:=diff(errsq,c3)=0:
> eq4:=diff(errsq,c4)=0:
> eq5:=diff(errsq,c5)=0:
> eq6:=diff(errsq,c6)=0:
> eq7:=diff(errsq,c7)=0:
> eq8:=diff(errsq,c8)=0:
> eq9:=diff(errsq,c9)=0:
> coefs:=solve({eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq9},{c1,c2,c3,c4,c5,
,c6,c7,c8,c9}):

```

```

*****
Determine the interpolating polynomial (forcefcn);
*****

```

```

> poly:=(c1+c2*S+c3*S^2+c4*CR+c5*W+c6*W^2+c7*L+c8*PR+c9*PR^2):
> forcefcn:=subs(coefs,poly);
forcefcn := -6.205488273 - 1.134335218 S + .1238142183 S^2 + 4.831517572 CR
- .2686768966 W + 3.707413519 W^2 + 1.364762117 L + 17.74800445 PR - 27.56074936 PR^2

```

```

*****
Carry out random testing with the following statement;
*****

```

```

> random:=simplify(12.925*2*Pi/4.448*(subs(S=1.625,CR=1.1,PR=0.42,W=
0.14,L=1.32,forcefcn)));
random := 36.91366405

```

```

*****
Compute total squared error error[i], fbar and fhat for calculating the
coeffecient of determination (rsq);
*****

```

```

>
for i from 1 to 104 do error[i]:=((abs(F[i]-mathforce[i]))^2) od:
> Fbar:=sum(F[k], 'k=1..104')/104:
> for i from 1 to 104 do

```

```

Fha[i]:=(c1+c2*S[i]+c3*S[i]^2+c4*CR[i]+c5*W[i]+c6*W[i]^2+c7*L[i]+c
8*PR[i]+c9*PR[i]^2) od:
> for i from 1 to 104 do Fhat[i]:=subs(coefs,Fha[i]) od:
> expvar:=sum((Fhat[k]-Fbar)^2,'k=1..104'):
> totvar:=sum((F[k]-Fbar)^2,'k=1..104'):

*****
Print coefficient of determination (rsq) for assessing interpolation function accuracy;
*****

> rsq:=expvar/totvar;

rsq := .9781760056

```

Appendix J

Maple Worksheet
Phase Two Calculations

Appendix J: Phase Two Maple Worksheet

Worksheet Objective: To develop the phase two interpolation force function; To compute its coefficient of determination; To compute interpolation function values for random testing; To compute all optimal configurations by using the penalty method.

Thesis: "Design of a Cost-Optimal Solenoid"

By: David R. Michael

University of Tennessee, Knoxville

Copyright August 1998: All Rights Reserved.

Filename: appen_J.mws

Insert Restart statement to clear memory

> restart;

Input Finite Element Results as Data Arrays;

Notes: Force values are normalized by dividing actual result in (lb.) by 12.925.

Stroke Position values are normalized by dividing actual value in (inches) by 0.08.

All other data is not normalized and has units of inches

```
> F:=array(1..104,1..391,1..142,.865,.602,1.285,1.054,.899,.655,1.342
,1.144,.9,.651,1.388,1.121,.918,.631,1.368,1.06,.843,.558,1.302,.9
94,.764,.511,1.226,.924,.712,.461,1.28,1.089,.841,.591,1.357,1.125
,.856,.597,1.381,1.137,.861,.6,1.397,1.147,.868,.603,1.4,1.15,.87,
.604,1.401,1.152,.87,.604,.889,.883,.896,.896,.739,.688,.902,.903,
.922,.849,.77,.713,1.005,.784,.54,.236,1.213,.97,.706,.328,1.513,1
.311,1.001,.531,1.655,1.438,1.145,.628,1.77,1.542,1.275,.73,.89,.6
69,.453,.287,1.134,.913,.66,.438,1.594,1.355,1.065,.769,1.778,1.54
,1.252,.934,1.949,1.708,1.421,1.099));
```

```
> S:=array(1..104,1..5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,
.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,
.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,
.5,.75,1,1.375,1,1,1,1,1,1,1,1,1,1,.5,.75,1,1.375,.5,.75,1,1.37
5,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,
.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375,.5,.75,1,1.375));
```

```

> PR:=array(1..104,[.35,.35,.35,.35,.298,.298,.298,.298,.315,.315,.3
15,.315,.333,.333,.333,.333,.368,.368,.368,.368,.385,.385,.385,.38
5,.403,.403,.403,.403,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.298,.315,.333
,.368,.385,.403,.298,.315,.333,.368,.385,.403,.35,.35,.35,.35,.35,
.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
5,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,.35,
.35,.35]):
> W:=array(1..104,[.105,.105,.105,.105,.105,.105,.105,.105,.105,.105
,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
.105,.105,.105,.105,.105,.074,.074,.074,.074,.084,.084,.084,.084,.
095,.095,.095,.095,.116,.116,.116,.116,.126,.126,.126,.126,.137,.1
37,.137,.137,.074,.074,.074,.074,.074,.074,.137,.137,.137,.137,.13
7,.137,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105
,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,.105,
.105,.105]):
> L:=array(1..104,[.962,.962,.962,.962,.962,.962,.962,.962,.962,.962
,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,
.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.9
62,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.96
2,.962,.77,.77,.77,.77,.866,.866,.866,.866,1.058,1.058,1.058,1.058
,1.154,1.154,1.154,1.154,1.251,1.251,1.251,1.251,.962,.962,.962,.9
62,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.962,.96
2,.962,.962,.962]):
> CR:=array(1..104,[.818,.813,.813,.813,.818,.813,.813,.813,.818,.81
3,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813
,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,
.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.
813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.8
13,.813,.818,.813,.813,.813,.818,.813,.813,.813,.818,.813,.813,.81
3,.818,.813,.813,.813,.818,.813,.813,.813,.731,.731,.731,.731,.772
,.772,.772,.772,.853,.853,.853,.853,.894,.894,.894,.894,.934,.934,
.934,.934]):

```

```

*****
Carry out least squares summation and solve for each unknown coefficient;
*****

```

```

> errsq:=sum(' (F[k]-1.*c1-1.*c2*S[k]-1.*c3*S[k]^2-1.*c4*CR[k]-1.*c5*
CR[k]^2-1.*c6*W[k]-1.*c7*W[k]^2-c8*L[k]-c9*L[k]^2-c10*PR[k]-c11*PR
[k]^2)^2','k'=1..104):
> eq1:=diff(errsq,c1)=0:

```

```
[ > eq2:=diff(errsq,c2)=0:
[ > eq3:=diff(errsq,c3)=0:
[ > eq4:=diff(errsq,c4)=0:
[ > eq5:=diff(errsq,c5)=0:
[ > eq6:=diff(errsq,c6)=0:
[ > eq7:=diff(errsq,c7)=0:
[ > eq8:=diff(errsq,c8)=0:
[ > eq9:=diff(errsq,c9)=0:
[ > eq10:=diff(errsq,c10)=0:
[ > eq11:=diff(errsq,c11)=0:
[ > coefs:=solve({eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8,eq9,eq10,eq11},{c1,c
[ 2,c3,c4,c5,c6,c7,c8,c9,c10,c11}):
```

```
*****
Determine the interpolating polynomial (forcefcn);
*****
```

```
> poly:=(c1+c2*S+c3*S^2+c4*CR+c5*CR^2+c6*W+c7*W^2+c8*L+c9*L^2+c10*PR
+c11*PR^2):
> forcefcn:=subs(coefs,poly);
forcefcn := -9.719691699 - 1.116166415 S + .1145743029 S^2 + 7.689014988 CR
- 1.733383014 CR^2 + 2.729532778 W - 10.51754244 W^2 + 4.381312471 L - 1.45911433 L^2
+ 21.46218599 PR - 32.86225422 PR^2
```

```
*****
Carry out random testing with the following statement;
*****
```

```
> random:=simplify(12.925*2*Pi/4.448*(subs(S=1.625,CR=1.1,PR=0.42,W=
0.14,L=1.32,forcefcn)));
```

```
random := 31.66832027
```

```
*****
Compute total squared error error[i], fbar and fhat for calculating the
coefficient of determination (rsq):
*****
```

```
>
for i from 1 to 104 do error[i]:=((abs(F[i]-mathforce[i])^2) od;
```

```

> Fbar:=sum(F[k], 'k=1..104')/104:

> for i from 1 to 104 do
  Fha[i]:=(c1+c2*S[i]+c3*S[i]^2+c4*CR[i]+c5*CR[i]^2+c6*W[i]+c7*W[i]^
  2+c8*L[i]+c9*L[i]^2+c10*PR[i]+c11*PR[i]^2) od:

> for i from 1 to 104 do Fhat[i]:=subs(coefs,Fha[i]) od:

> expvar:=sum((Fhat[k]-Fbar)^2, 'k=1..104'):

> totvar:=sum((F[k]-Fbar)^2, 'k=1..104'):

*****
Print coefficient of determination (rsq) for assessing interpolation function accuracy;
*****

> rsq:=expvar/totvar;

rsq := .9815739036

*****
Print solcost to be used in penalty method;
*****

> solcost :=
1.110000000-.3911282854*CR^2-3.392920066*CR*W*L+1.277999892*CR*W+1
.696460034*W^2*L-.6389999458*W^2+.2000000000e-1*CR-2.042035226*PR^
2*L+.5871636670*PR^2+.1000000000*PR+2.827433389*L*CR^2;
solcost := 1.110000000 - .3911282854 CR^2 - 3.392920066 CR W L + 1.277999892 CR W
+ 1.696460034 W^2 L - .6389999458 W^2 + .02000000000 CR - 2.042035226 PR^2 L
+ .5871636670 PR^2 + .1000000000 PR + 2.827433389 L CR^2

*****
User input minimum force requirement (lb.) and corresponding stroke position (in.);
*****

> minimumforce:=(13):

> strokeposition:=(0.08):

```

```
*****
Setup penalty method equations;
*****
```

```
> Mforce:=eval(minimumforce/12.925);
> force:=eval(subs(S=strokeposition,forcefcn));
> penalty1:=solcost-lambda*(Mforce-force);
> penalty:=simplify(expand(penalty1));
penalty:=1.110000000-.3911282854 CR2-3.392920066 CR W L+1.277999892 CR W
+1.696460034 W2 L-.6389999458 W2 +.02000000000 CR-2.042035226 PR2 L
+.5871636670 PR2 +.1000000000 PR+2.827433389 L CR2-10.81405444 λ
+7.689014988 λ CR-1.733383014 λ CR2+2.729532778 λ W-10.51754244 λ W2
+4.381312471 λ L-1.475911433 λ L2+21.46218599 λ PR-32.86225422 λ PR2
```

```
*****
Differentiate penalty equation (penalty) with respect to each of 5 variables;
*****
```

```
> peneq1:=diff(penalty,CR)=0;
peneq1:=-.7822565708 CR-3.392920066 W L+1.277999892 W+.02000000000
+5.654866778 L CR+7.689014988 λ-3.466766028 λ CR=0
> peneq2:=diff(penalty,PR)=0;
peneq2:=
-4.084070452 PR L+1.174327334 PR+.1000000000+21.46218599 λ-65.72450844 λ PR=0
> peneq3:=diff(penalty,L)=0;
peneq3:=-3.392920066 CR W+1.696460034 W2-2.042035226 PR2+2.827433389 CR2
+4.381312471 λ-2.951822866 λ L=0
> peneq4:=diff(penalty,W)=0;
peneq4:=-3.392920066 L CR+1.277999892 CR+3.392920068 W L-1.277999892 W
+2.729532778 λ-21.03508488 λ W=0
> peneq5:=diff(penalty,lambda)=0;
peneq5:=-10.81405444+7.689014988 CR-1.733383014 CR2+2.729532778 W
-10.51754244 W2+4.381312471 L-1.475911433 L2+21.46218599 PR-32.86225422 PR2=
0
```

```
*****
Solve for all real optimal roots and check to see if analysis ranges are exceeded;
```

Note: None of these real optimum roots (optroots) are valid if one or more do not fall within the original finite analysis range.

```
> optroots:=fsolve({peneq1,peneq2,peneq3,peneq4,peneq5},{CR,PR,L,W,lambda});
```

```
optroots:= {
```

```
CR=.7809808537, L=.6533352106, PR=.3412367752, W=.1917143173, lambda=-.4244593124 }
```

Print optimum force value (lb.) and optimal solenoid cost (\$) as determined by all real optimum roots;

```
> subs(optroots,force)*12.925;
```

```
12.99999996
```

```
> subs(optroots,solcost);
```

```
1.837607071
```

Appendix K

Fortran Code

Phase Two Search Algorithm

```

* David Michael
* Copyright August 1998
* THESIS: Design of a Cost Optimal Solenoid
*
* Purpose: To find the optimum variable set with respect to cost for a given
*          force requirement and stroke distance
*
* .....

```

PROGRAM SEARCH

```

* .....
*
* Variables: L, LLow, LHi, dL
*            Cr, CrLow, CrHi, dCr
*            Pr, PrLow, PrHi, dPr
*            W, WLow, WHi, dW
*            S, force, forcemin
*            cost, costmin
*
* .....

```

```

IMPLICIT REAL (a-h,i,o-z)

```

```

READ*,WLow, WHi, PrLow, PrHi
READ*,LLow, LHi, CrLow, CrHi
READ*,stroke, forcemin

```

```

flag = 0.0

```

```

Nmax = 10000

```

```

costmin = 10.00

```

```

S = stroke/0.08

```

```

dCr = .01

```

```

dPr = .01

```

```

dL = .01

```

```

dW = .01

```

```

Cr = CrLow-dCr

```

```

2      DO Isteps = 0, Nmax

```

```

      W = WLow - dW

```

```

      L = LLow - dL

```

```

      Pr = PrLow - dPr

```

```

      Cr = Cr + dCr

```

```

3      DO Nsteps = 0, Nmax

```

```

      W = WLow - dW

```

```

      Pr = PrLow-dPr

```

```

      L = L + dL

```

```

10      DO Ksteps = 0, Nmax
          W = WLow - dW
          Pr = Pr + dPr

          DO Jsteps = 0, Nmax
              W = W + dW

              IF (W .GT. WHi) THEN
                  GOTO 10
              END IF

              IF (Pr .GT. PrHi) THEN
                  GOTO 3
              END IF

              IF (L .GT. LHi) THEN
                  GOTO 2
              END IF

              IF (Cr .GT. CrHi) THEN
                  GOTO 50
              END IF

              CALL FORCEVAL(force,S,Pr,W,L,Cr)
              CALL COSTVAL(cost,Pr,W,L,Cr)

              IF (force .GT. forcemin) THEN
                  flag = 1.0

                  If (cost .LT. costmin) THEN
                      Propt = Pr
                      Wopt = W
                      Lopt = L
                      Cropt = Cr
                      forceopt = force
                      costmin = cost
                  END IF
              END IF
          END DO
      END DO
  END DO
  END DO
  END DO

```

```
50      IF (flag.LT..99) THEN
```

```
        PRINT*, '*****'
        PRINT*, 'Input:'
        PRINT*, 'Minimum Force =', forcemin
        PRINT*, 'Stroke Position =', stroke
        PRINT*, ' '
        PRINT*, ' '
        PRINT*, 'No Configurations Will Meet This Requirement'
        PRINT*, ' '
        PRINT*, ' '
        PRINT*, '*****'
```

```
        STOP
```

```
    END IF
```

```
    PRINT*, ' '
    PRINT*, 'Search Complete...'
    PRINT*, ' '
    PRINT*, ' '
    PRINT*, '*****'
    PRINT*, ' '
    PRINT*, 'Input:'
    PRINT*, ' '
    PRINT*, 'Minimum Force =', forcemin
    PRINT*, 'Stroke Position =', stroke
    PRINT*, ' '
    PRINT*, 'Optimal Configuration:'
    PRINT*, ' '
    PRINT*, 'Pr =', Propt
    PRINT*, 'W =', Wopt
    PRINT*, 'L =', Lopt
    PRINT*, 'Cr =', Cropt
    PRINT*, ' '
    PRINT*, 'Output: '
    PRINT*, ' '
    PRINT*, 'FORCE =', forceopt
    PRINT*, 'COST =', costmin
    PRINT*, ' '
    PRINT*, '*****'
```

```
END
```

```
*****
* Subroutine to Calculate the Force
```

```
SUBROUTINE FORCEVAL(force,S,Pr,W,L,Cr)
```

```
IMPLICIT REAL (a-h,i,o-z)
```

```

      force = (-9.721181251-1.116157118*S
m      +.1145696023*S**2+21.46438627*Pr-32.86539325*Pr**2
m      +2.729950693*W-10.51952219*W**2-4.381362201*L
m      -1.475938732*L**2+7.691547587*Cr-1.734896770*Cr**2)
m      *12.925*2*3.1415926/4.448
```

RETURN

END

* Subroutine to Calculate the Cost

SUBROUTINE COSTVAL(cost,Pr,W,L,Cr)

IMPLICIT REAL (a-h,i,o-z)

cost = 1.111100000-.3911282854*CR**2-3.392920066*CR*W*L
m +1.277999892*CR*W+1.696460034*W**2*L-.6389999458*W**2
m +.2000000000e-1*CR-2.042035226*PR**2*L+.5871636670*PR**2
m +.1000000000*PR+2.827433389*L*CR**2

RETURN

END

* Input File
* Filename: dat

0.074 .137 .298 .403 \ WLow, WHi, PLOW, RHi
0.77 1.251 0.731 0.934 \ LLow, LHi, PLOW, PHi
0.07 25.0 \ stroke, forcemin

VITA

David R. Michael was born in Charleston, WV on June 9, 1974. He attended the Kanawha County public school system, where he graduated from Capital High School in June, 1992. He entered the University of Tennessee, Knoxville in August, 1992 and graduated with a Bachelor of Science in Engineering Science and Mechanics in May, 1996. After spending a summer as a product engineer for Hydro-Force Corporation, he returned to the University of Tennessee in August 1996 to begin work in the Engineering Science and Mechanics graduate program. As a graduate student, he was a Basic Engineering Graphics instructor , Engineering Fundamentals Graduate Assistant, and was a member of the Basic Engineering Renovation Team. His academic focus was in the field of computational solid mechanics and finite element method applications in design optimization.